Community Detection in Multi-Layer Networks

Oksana Pichugina

Mathematics and Statistics

A thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Science in Mathematics

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

Abstract

In the scope of the current thesis we review and analyse networks that are formed by nodes with several attributes. We suppose that different layers of communities are embedded in such networks, besides each of the layers is connected with nodes' attributes. For example, examine one of a variety of online social networks: an user participates in a plurality of different groups/communities – schoolfellows, colleagues, clients, etc. We introduce a detection algorithm for the above-mentioned communities. Normally the result of the detection is the community supplemented just by the most dominant attribute, disregarding others. We propose an algorithm that bypasses dominant communities and detects communities which are formed by other nodes' attributes. We also review formation models of the attributed networks and present a Human Communication Network (HCN) model. We introduce a High School Texting Network (HSTN) and examine our methods for that network.

Acknowledgement

I thank my research supervisor Dr. Babak Farzad for his help, patience and support through this Master's thesis. I am grateful to Dr. Omar Kihel and Dr. Henryk Fuks for letting me have them as my thesis committee members. I am grateful to Brock University for providing me financial support. I thank Stephanie Noel for collecting data for the High School Texting Network.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Network Analysis is an area of research that was studied intensively lately. Researches investigate structural characteristics of different networks, network formation models and many other related questions. Among a variety of networks, social networks, which reflect a diversity of people relationships, are a priority.

Why the investigation of social networks is so important? First of all, it helps to understand how our world is organized, what place each of us takes in it, how this situation affects us and how the knowledge can be used to achieve our goals. In many types of networks and in social networks particularly exist observable and tightly bound groups of elements called *communities*. How and why communities arise is an interesting and important question. It is also important to analyse possible ways of *community detection* (CD). There is a whole domain of Network Analysis that studies this. CD researchers

develop *CD Algorithms* (CDAs) with a purpose to extract communities fast and qualitatively. The goal can be achieved by using maximum of the available information, but it is a challenging task because nowadays networks can be immeasurably huge.

1.1 Background and Motivations

Imagine hypothetically that our goal is to study a community structure of the whole mankind social network. Suppose we have the absolute information about its participants, their contacts, and, the most important, the power of these contacts. Imagine we have a supercomputer and an ideal CD Algorithm, CDA. The result of the CD is fully predictable – it would be a division by families because normally people spend most of the time with their families and because family ties are very strong. However, when we set up the experiment we expected to get something we did not know before. Our expectations were to see the network from different sides, for example to see the relationship between friends or between colleagues.

Our assumption is that it would be possible by allocating and then deleting the dominant subnetwork of the family relationships that creates the first layer of the global network. Conducting CD in the remaining network would demonstrate a new community structure. In this case the result is not obvious – for some people the next important thing is a friendship, for others – a hobby or a job and so on. Nevertheless, it makes sense to detect the

second most important factor, say a friendship, detect in the global network a friendship subnetwork and remove it from the consideration. Thus, other new layers of the global network can be detected.

In terms of Network Analysis, nodes of the global network are people and edges are their relationships. Each person has a number of characteristics: belonging to a certain family, to circles of friends and classmates, a list of hobbies and interests, etc. These characteristics are represented as *attributes* of the nodes. These attributes underlie a network built on similarity of the node attributes, which is called an *association network*. A network with nodes' or edges' attributes is called an *attributed network*.

Due to the variety of characteristics, nodes in social networks are heterogeneous, as well as edges. For the described multi-layer community detection the heterogeneity, in particular presence of node attributes, is crucial. We use attributes of nodes to detect which one of them is the main reason of the particular community structure. Then we use an edge attribute weight to extract a subnetwork of the detected layer.

In the scope of the current thesis we review the following subjects related to attributed network analysis:

- **Problem 1**. Community detection in attributed networks that combines a network structure accumulated in edges data and node attribute information.

- **Problem 2**. Reconstruction of the missing nodes' and edges' attributes.

- **Problem 3**. Interpretation of the CD results.
- **Problem 4**. Attributed networks formation models.

1.2 Organization of the Thesis

The thesis is dedicated to Problems 1-4 and is organized as follows. Chapter 2 introduces terminologies used throughout the thesis, describes properties of social networks and community detection algorithms. Chapter 3 contains the information that is already known and will be developed in the scope of the thesis – approaches for dealing with association and attributed networks, relevant CDAs and the Analytic Hierarchy Process. The latter is used to construct an aggregated network that combines the structural and attribute information of attributed networks. Chapter 4 is dedicated to attributed networks analysis and contains the following theoretical contributions: a) the aggregated network utilizes available attributed network information based on priorities of its components; b) the aggregated network is decomposed into sub-networks corresponding to individual node attributes and the decomposition is used for Problems 1-3, namely in the Multi-Layer Community Detection algorithm (MLCD algorithm) and in the Multi-Layer Node Attribute Inference algorithm (MLNI algorithm); c) in the scope of Problem 4 a number of the attributed network formation models are presented, in particular a Human Communication Network model (HCN model). A High School Texting Network (HSTN) is also introduced in Chapter 4 while analysis of the network

is presented in Chapter 5. Chapter 5 is also dedicated to an experimental part of the research: simulation of attributed networks, implementation of the MLCD and the MLNI algorithms, and interpretation of the HSTN analysis results.

Chapter 2

Preliminaries

2.1 Definitions and Notations

A *network* is a collection of connected objects. It can be mathematically represented by a graph with nodes (vertices) and edges (links) corresponding to the objects and connections between them. Therefore we formulate some definitions common in Graph Theory in terms of Network Analysis.

Let $G = (V, E)$ be a simple weighted network on n vertices and m edges ($|V| = n, |E| = m$).

The network's *adjacency matrix* is a matrix A of order n such that

$$a_{ij} = \begin{cases} 1, & \text{if } v_i, v_j \text{ are adjacent } (v_i \leftrightarrow v_j), \\ 0, & \text{otherwise} \end{cases} \qquad (v_i, v_j \in V).$$

The network's *weighted adjacency matrix* (WAM) is a matrix A^w of weights of its edges.

The *node degree* d_i of a vertex $v_i \in V$ is the number of its incident edges, $d_i = |N_i|$, where $N_i = \{u \in V : u \leftrightarrow v_i\}$.

The *node strength* s_i of a vertex $v_i \in V$ is a sum of weights of its incident edges.

In terms of A and A^w, the node degree and the node strength can be represented as follows:

$$d_i = \sum_j a_{ij},\ s_i = \sum_j a^w_{ij}.$$

Let G and H be weighted networks of the same order.

A *sum*, $G + H$, or a *difference*, $G - H$, of networks G and H [LWBC04] is a network of the same order with the WAM equal to the sum or difference of the weighted adjacent matrices (WAMs) of G and H.

This definition can be generalised into linear operations on weighted networks as following:

Let $\{G_i\}_{i \in J_K}$ be a set of weighted networks with the same node set ($V(G_i) = V, i \in J_K = \{1, ..., K\}$) and WAMs $\{A_i\}_{i \in J_K}$.

A *linear combination of the networks* $\{G_i\}_i$ is a network with a WAM given by the corresponding linear combination of $\{A_i\}_i$.

So, G' is the linear combination of the networks $\{G_i\}_i$:

$$G' = \sum_{i \in J_K} \alpha_i \cdot G_i, \tag{2.1}$$

if $\exists \alpha = (\alpha_i)_i \in R^K$ such that the WAM A' of G' is equal to $A' = \sum_{i \in J_K} \alpha_i \cdot A_i$. Introduce $||A|| = \sum_{i \in J_m, j \in J_n} a_{ij}$ - the *norm of a matrix* $A \in R^{m \times n}$ - and $||b|| = \sum_{i \in J_n} b_i$ - the *norm of a vector* $b \in R^n$.

If $\alpha \in R^K : \alpha \geq 0$, $||\alpha|| = 1$ then the linear combination (2.1) of $\{G_i\}_{i \in J_K}$ is called the *weighted network sum.*

The sum $\omega(G)$ of E-weights is a *weight* of the network G:

$$\omega(G) = \sum_{i,j \in J_n} a_{ij}^w = ||A^w||. \tag{2.2}$$

A network G of the weight 1 is called *normalized network*:

$$\omega(G) = 1. \tag{2.3}$$

A WAM of a normalized network we denote by $\overline{w} = (w_{ij})_{ij}$. Then $\forall G : ||G|| \neq 0$ we can correspond the normalized network $G' = \frac{G}{\omega(G)}$ with the WAM

$$\overline{w} = \frac{A^w}{||A^w||}. \tag{2.4}$$

The *disjoint union* $G = \bigcup\limits_{i \in J_k} G_i$ of networks $\{G_i\}_{i \in J_K}$ with disjoint nodes sets $\{V_i\}_{i \in J_K}$ and edge sets $\{E_i\}_{i \in J_K}$ is a network with nodes $V = \bigcup\limits_{i \in J_K} V_i$ and edges $E = \bigcup\limits_{i \in J_K} E_i$.

A network partition is a division of its nodes

$$\mathcal{C} = \Pi(G) = \{C_l\}_{l \in J_L} : \tag{2.5}$$

$$\bigcup_{l=1}^{L} C_l = V, \tag{2.6}$$

where node clusters $C_l \in V, l \in J_L$, are pairwise disjoint.

A network cover is a division (2.5) of the network nodes satisfying (2.6).

Let [SMM14] $E(C, C')$ denotes the set of edges between nodes' clusters $C, C' \subseteq V$: $E(C, C') = \{\{u, v\} \in E : u \in C, v \in C'\}$, $\omega(C, C') = \sum\limits_{\{i,j\} \in E(C,C')} a_{ij}^w$. Respectively, if $E(C) = E(C, C)$ then $\omega(C) = \omega(C, C)$.

Let $C \subseteq V$, $G[C]$ be an *induced G-subnetwork* by C:

$$G[C] = (C, E(C)),\ n(C) = |C|.$$

In these notations there are defined a set $\{G[C_l]\}_{l \in J_L}$ of subnetworks induced by each cluster C_l in the partition (2.5): $n(C_l) = |C_l|$, $\sum_l n(C_l) = n$.

The *internal* or *external degree of vertex* [Das14] $u \in C \subseteq V$ is the number of edges connecting u to other vertices of C, C-inter-cluster edges, or to the rest of the network, C-external edges, respectively:

$$d_u^{int,C} = |E(u, C)|,\ d_u^{ext,C} |E(u, \overline{C})|,\ \overline{C} = V \backslash C.$$

The sets $E(C)$, $E(C, \overline{C})$ are *C-intra-cluster edge set* and *C-inter-cluster edge set*, respectively.

The sum of the internal or external degrees of C-vertices is the *internal* or *external degree of the cluster* C, respectively: $d^{int,C} = \sum_{u \in C} d_u^{int,C} = 2 \cdot |E(C)|$, $d^{ext,C} = \sum_{u \in C} d_u^{ext,C} = |E(C, \overline{C})|$.

The *total degree of the cluster* C is the sum of its internal and external degrees: $d^C = d^{int,C} + d^{ext,C} = 2 \cdot |E(C)| + |E(C, \overline{C})|$.

Using edge weights we can similarly define the *intra-cluster weight* $\omega(C)$, *C-cut weights* $\omega(C, \overline{C})$ and *total C-cluster weight* $\omega(C) + \omega(C, \overline{C})$.

For the partition (2.5) $E(\mathcal{C}) = \underset{l \in J_L}{\cup} E(C_l)$ is the *set of intra-clusters edges* and $E(\overline{\mathcal{C}}) = E \backslash C$ is the *set of inter-clusters edges* of the partition $\mathcal{C}$.

In these notations the *internal degree* and the *internal weight of the partition* $\mathcal{C}$ are:

$$d^{int}(\mathcal{C}) = \sum_{C \in \mathcal{C}} d^C, \; \omega^{int}(\mathcal{C}) = \sum_{C \in \mathcal{C}} \omega(C); \tag{2.7}$$

the *degree* and the *weight of the partition* $\mathcal{C}$ are:

$$d(\mathcal{C}) = d(G) - d(\overline{\mathcal{C}}), \; \omega(\mathcal{C}) = \omega(G) - \omega(\overline{\mathcal{C}}). \tag{2.8}$$

The ratio between the number of C-intra-cluster edges and the number of all possible internal edges of the nodes' cluster C is called the *intra-cluster density* of the subnetwork $G[C]$:

$$\delta^{int,C} = \frac{2|E(C)|}{|C|(|C|-1)}. \tag{2.9}$$

The ratio between the number of C-inter-cluster edges and the number of all possible C-external edges is called the *inter-cluster density* of $G[C]$:

$$\delta^{ext,C} = \frac{|E(C, \overline{C})|}{|C| \cdot |\overline{C}|}. \tag{2.10}$$

The *network density* is the ratio between the number of edges and the number of all possible edges in the network: $\delta = \frac{2m}{n(n-1)}$.

The *local clustering coefficient* CC_i for a vertex $v_i \in V$ is the proportion of links between the vertices within its neighbourhood (d'_i) divided by the number of links that could possibly exist between them: $CC_i = \frac{2d'_i}{d_i(d_i-1)}$. The local clustering coefficient can be interpreted as the probability that two of one's friends are also friends themselves.

The *average clustering coefficient* (ACC, transitivity) $\overline{CC}$ is the mean of the local clustering coefficients of all the vertices:

$$\overline{CC} = \frac{1}{n}\sum_i CC_i. \tag{2.11}$$

The *shortest path length* $l(u, v)$ between vertices $u, v \in V$ is the minimal sum of weights of its constituent edges.

The *average shortest path length* (ASPL) $\bar{l}$ is the mean of the shortest path lengths averaged over all pairs of nodes:

$$\bar{l} = \frac{1}{m}\sum_{\{u,v\}\in E} l(u, v). \tag{2.12}$$

The *distance* $dist(u, v)$ between vertices $u, v \in G$ is the minimal number of its constituent edges.

The *diameter* of a network G is maximal distance between its nodes: $Diam(G) = \max_{\{u,v\}\in V} dist(u, v)$.

The *conductance* $\Phi(C)$ of a cluster C in a weighted network G is defined [LLM10] as

$$\phi(C) = \frac{\omega(C, \overline{C})}{min(\omega(C), \omega(\overline{C}))}. \tag{2.13}$$

For unweighted networks (2.13) becomes the fraction of the number of its inter-cluster edges and the minimum of degrees of C and $\overline{C}$ [YL12],[For10]:

$$\phi(C) = \frac{|E(C, \overline{C})|}{min(d^C, d^{\overline{C}})}.$$

2.2 Properties of Social Networks with Focus on Communities

Social networks are networks of people relationships. In the corresponding network graphs people (the social networks users) can be represented by nodes and the users' relationships - by edges. But all people are different and their relationships too. This people variety can be represented by attributes of the nodes. The same the diversity of relationships can be depicted by attributes of the edges.

So, any social network can be represented as follows:

Definition 1. *[LZXC14] A social network is a hybrid graph, which is represented in the form:*

$$G = (V, E, \Lambda, \Lambda'), \tag{2.14}$$

where V is the set of nodes (the social network's users), E is the set of edges

(these users relationships), Λ *and* Λ' *contain an information about attributes related to each node* $v \in V$ *and each edge* $\{u, v\} \in E$*, respectively.*

Since social networks deal with people relationships, they differ from other networks, such as technological, informational, biological, biochemical, etc. Social networks have specific properties, because they are formed by people's necessity to communicate: it is common that social networks are sparse, have a small diameter and a high clustering coefficient. There is a significant number of sociable users with many relationships in social networks, who are represented by high degree nodes called *hubs*. At the same time there is a considerable number of unsociable users having a few connections and representing by low degree vertices in the networks. This is reflected in a "heavy" tail node degree distribution comparing to a typical exponential distribution. One more important feature of social networks is existence of dense subnetworks called communities.
Most of the listed properties are united in "small-world networks" and "scale-free networks" concepts [New03], [Kol09], [WS98], [AB02].

Definition 2. *[WS98] A small-world network is a network* G*, where typical distance* $dist(u, v)$*, between two randomly chosen nodes* $u, v \in G$*, grows proportionally to the logarithm of the number of nodes in the network, implying that*

$$dist(u, v) \propto log\, n. \tag{2.15}$$

A list of properties of small-world networks:

1. **main** small-world networks features are high $\overline{CC}$ and small $\bar{l}$;

2. **additional** small-world networks properties, which follow from the main ones are: a) existence of numerous cliques and near-cliques as a result of high $\overline{CC}$; b) connection by at least one short path of most of node pairs as a consequence of small $\bar{l}$;

A *scale-free network* [AB02] is a network whose degree distribution follows a power law implying that probabilities $p(k)$ of nodes with k connections to the rest ones can be approximated as follows:

$$\exists \gamma \in R : \; p(k) = P(d_v = k) \sim k^{-\gamma}, \; v \in V. \tag{2.16}$$

(2.16) is reflected, in particular, in overabundance of hubs and low-degree nodes.

Due to all these specific properties, social networks are in the area of interest of many researches [GN02], [Kol09], [CA12]. For example in social networks modelling researches mainly focus on reproducing small-world properties (2.15) and scale-free networks (2.16), but still no satisfactory solution to simulate all the listed social networks properties is found [WS98],[AB02],[Kol09]. Another area of interest is study of nature of dense subnetworks called communities. Their extracting from networks is called community detection (CD).

Detecting communities and studying their nature are very important for any kind of networks (see Section 2.4), because it offers insight into many of the networks phenomena. For social networks especially these issues are relevant

because: a) nowadays social networks are high-scale, and even their clear visual representation needs merging similar vertices in groups; b) sociologists investigate reasons for formatting tightly knit groups of people, etc.

Communities can overlap or not. If overlapping is acceptable, then solution of the CD problem is a cover of V, otherwise it is a node partition (see (2.5)). Typically, CD refers to nodes' partitions. We will consider two types of the partitions - node partitions into communities (obteined as a result of implementing a CDA) and node partitions into clusters related to different values of a particular node attribute. Then node covers are formed from these partitions (see Section 3.2.3).

For the clarification we use the following notation for a partition (2.5) into communities:

$$\mathcal{C}^* = \{C_{l^*}\}_{l^* \in J_{L^*}} \text{ is a } G\text{-partition into communities,} \tag{2.17}$$

$$|C_{l^*}| = |G[C_{l^*}]| = n_{l^*},\ l^* \in J_{L^*},\ \sum_{l^* \in J_{L^*}} n_{l^*} = n. \tag{2.18}$$

2.3 Problem Statements

Let us consider a network G. Suppose that in addition to the basic information about the vertex set V and the edge set E we have some extra information about nodes features, edges characteristics and peculiarities of the whole network. These additional characteristics are called attributes, the procedure

of their complementing is *decoration* of the network [BGL05], [Kol09] resulted in creation of an attributed network [ZCY10]. For example, if G is a traffic network, where nodes are cities and edges are roads, then the nodes can be decorated by "population", "capacity", etc., and edges can be accompanied by "cost", "capacity", etc. At the same time the whole network can be also decorated, for instance by its type "technological network", more specifically, it is a transportation network.

How such kind of additional information can help us in understanding a particular network and especially its community structure? Having information about network attributes we may expect that communities, detected by community detection algorithms (CDAs), are related to one node's attribute (ATN) or to a combination of these attributes. Here Question 1 appears "How to link the community structure with node attributes and justify the choice?" (see Section 4.3.2). Notice that the assigning node attributes to communities is also network decoration (of communities).

Let us introduce some notations for a network (2.14), which nodes' and edges' attributes take finite number of values. Then let $\overline{AT}^n = (AT^{nk})_{k\in J_K}$ be a tuple of discrete nodes' attributes (ATNs) taking values $\overline{at}^n = (at^{nk})_{k\in J_K} = (\{at_l^{nk}\}_{l\in J_{L_k}})_{k\in J_K}$ (ATNVs); $\overline{AT}^e = (AT^{ek})_{k\in J_{K'}}$ be a tuple of edges' attributes (ATEs) with values $\overline{at}^e = (at^{ek})_{k\in J_{K'}} = (\{at_l^{ek}\}_{l\in J_{L'_k}})_{k\in J_{K'}}$ (ATEVs); $\overline{AT}^{C^*} = (AT^{C^*k})_k$ be a tuple of communities' attributes with $\overline{at}^{C^*} = (at^{C^*k})_k = (\{at_l^{C^*k}\}_l)_k$-values, and $\overline{ATG} = (ATG^k)_k$ be a tuple of the network

G attributes. Let

$$\overline{a}_i = (a_i^k)_{k\in J_K},\ \overline{a}'_{\{i,j\}} = (a'^k_{\{i,j\}})_{k\in J_{K'}}\ (a_i^k \in at^{nk},\ a'^k_{\{i,j\}} \in at^{ek})_k \tag{2.19}$$

denote tuples of ATNVs of a node v_i and ATEVs of an edge $\{v_i, v_j\} \in E$, respectively ($v_i, v_j \in V$). In terms of (2.19), Λ and Λ' in (2.14) can be represented as follows:

$$\Lambda = (\overline{a}_i)_{i\in J_n},\ \Lambda' = (\overline{a}'_{\{i,j\}})_{\{i,j\}\in E}. \tag{2.20}$$

A network with partially missing node or edge attributes is called [LKY+12] an *incomplete information attributed network* (IIAN). Analogically to this the network with present all as nodes' as edges's attributes we call a *complete information attributed network* (CIAN).

By these notations decoration of communities can mean an assignment of a certain $at^{C_{l^*}k} \in \overline{at}^n$ or $at^{C_{l^*}k} \in \overline{at}^e$ to a certain community $C_{l^*} \in \mathcal{C}^*$ (see (2.17)). On the other hand, it can imply deriving other properties from these communities, different from connection to a particular node or edge attribute, namely if $\exists k' : at^{C_{l^*}k'} \notin \overline{at}^n$, $at^{C_{l^*}k'} \notin \overline{at}^e$. For example, it can be a numerical characteristic of communities, information regarding communities nature and so on. The same is true for the whole network - the network's attributes can be derived from attributes of nodes, edges, or communities as well as from the network structure on the whole. For instance, analysis of the network may convince that it is a social, biological, technological network or it is just a

random graph. The derived type of the network is the result of its decoration by the attribute. Notice that depending on either partially or completely decorated by $\overline{at}^n$-vaues communities in $\mathcal{C}^*$ are, they also form an IIAN or a CIAN with nodes representing these communities and with edges, which weights equal to the total edge weight between the communities.

Since assigning attributes to communities can be also considered as recovering of missing attributes, we come to the next question (Question 2) regarding restoring missing network information. Recovering of missing network information may regard to any of its elements: a) edges information such as ATEs restoring or edges renovation; b) reconstruction of nodes and nodes' attributes (the last one implies a transformation of an IIAN into a CIAN); c) recovering the communities' and network' attributes. Notice that for making difference between Questions 1 and 2 we assume that the last one supposes renovation of attribute values exceptionally from sets $\overline{at}^n$, $\overline{at}^e$, for which the following is true $\overline{at}^{C^*} \subseteq \overline{at}^n \cup \overline{at}^e$. At the same time answering Question 1 we may obtain at^{C^*k} reflected by properties of specific communities, which are not typical as for nodes as for edges.

The same as to assign justified ATNVs to communities in $\mathcal{C}^*$ (see (2.17)), it is possible to attach $AT^{C^*k} \in \overline{AT}^n$ underlying the certain node partition if, for instance, for majority of the communities $at_l^{C^*k}$-values are assigned. The attribute AT^{C^*k} may be not uniquely defined, and it may be reflected in unstable CD results if, for instance, modularity values (see (2.22)) are similar in series of experiments, but partitions are very different. Suppose AT^{nk^*}

and $AT^{nk^{**}}$ ($k^* \neq k^{**}$) are independent node attributes equally underlying the community structure (2.17). It means that the structure is not uniquely defined due to existence of at least two strong node partitions, say $\mathcal{C}^*$ and $\mathcal{C}^{**}$, related to the attributes AT^{nk^*} and $AT^{nk^{**}}$, respectively. Clearly, it is reasonable to study these partitions individually. $\mathcal{C}^*$ and $\mathcal{C}^{**}$ form two overlapping nodes partitions located in two different levels, which we call network's *layers*. In real-world networks it is common to see existence of a dominant node attribute of CD-partitions (partitions into communities). If this node attribute is AT^{nk^*} and $\mathcal{C}^*$ is the corresponding dominant node partition, then the last one does not allow to obtain $\mathcal{C}^{**}$ and detect $AT^{nk^{**}}$ at all. It means that applying standard CD the valuable information about existence of the second partition related to $AT^{nk^{**}}$ is completely lost, and it comes out as a "noise" in the dominant partition. If the aim is to detect both, $\mathcal{C}^*$ and $\mathcal{C}^{**}$, as well as the rest of partitions related to ATNs, then it can be done only consecutively level by level. Unfortunately, traditional approaches to multilevel CD, like hierarchical clustering (see [RSM+02], [CMN08]), are not helpful because they suppose finding subcommunities of the next levels within communities of higher levels, but they do not suppose overlapping communities detection (see [PDFV05], [Gre10], [Li13]). At the same time overlapping CDAs do not allow us to use knowledge that we deal with a special cover, which is overlapping of several partitions, where each node belongs exactly to one community in each of these partitions. So, next question (Question 3) is "How to find such multi-layer partitions?". Finally, very

important question (Question 4) is "How attributed networks are formed, namely, why specific their links are created?".

These questions are directly related to our Problems 1-4 (see Chapter 1). Let us set up conformity between them:

- Question 1 is related to Problem 3. We expand the concept of nodes' and edges' attributes to attributes of communities and of the network as a whole. Assigning sets of attributes to communities allows us to interpret results of CD and say why these communities were formed and which specific features underlie them. For that a comparison of proportions of nodes with particular node attribute values in the derived communities and in the whole networks can be made (see Section 4.3.2). Conducting multi-layer CD we can obtain sets of communities attributes related to each of the obtained partitions;

- Question 2 is related to Problem 2. The problem can be named as "The decoration of networks by missing nodes' and edges' attribute values" and is related to the class of network inference (see Section 3.1). We will touch only node attributes reconstruction, more specifically renovation of missing ATNVs based on results of CD (see Section 4.3.2). Notice that our MLCD algorithm works only with weighted networks, that is why the edge weights' assignment is crucial part. Therefore we conduct one more kind of network decoration - we convert unweighted networks into weighted ones decorating their edges by weights (see Section 4.3.1).

- Question 3 is connected with Problem 1. We recommend iterative schemes of reducing the weights within attribute clusters of detected underlying node attributes of higher layers before CD of the next bottom layer (see Section 4.3.2). We apply the approach to the original network and to an accumulated network combined the last one with the corresponding association network;

- Question 4 concerns Problem 4 of modelling attributed networks. We assume that nodes are connected only due to their similarities accumulated in ATNVs, an element of randomness is present in the connections establishing, and an edge degree distribution depends on nature of the network. For example, weights of edges may depend on many reasons: a) the power of the edges incident nodes similarity; b) the power the incident nodes similarity to the rest of their neighbours; c) presence of restrictions on the number of connections or their intensity and so on. We present a number of such models, in particular, models of human communication networks (see Section 4.4.2).

2.4 Community Detection

Due to various applications of CD [GN02], [LF09], [New12], a lot of CDAs are well known (see surveys [For10], [Sch07]) and this area of Network Analysis is highly developed.

Nevertheless, many questions are still open. Communities identification is a

challenging task and there are several reasons for this:

- First of all, there is no unique definition of communities [For10], [Sch07], hence, there is no unique numerical measure Cr of community detection quality. The scoring function Cr can be based on the internal connectivity (Type A) or the external connectivity (Type B); it can be also a combination of the internal and external connectivities (Type C) or it can be based on a network model (Type D) [YL12]. Thus, if we aim to get a partition with high intra-cluster density (2.9) and low inter-cluster density (2.10) of detected communities, then two-criterion scoring function of C-type may be the following: $Cr = \sum_{C \in \mathcal{C}} \left(\delta^{int,C} - \delta^{ext,C} \right) \to max$ [For10]. Taking only one of these goals we get criteria of A-type and B-type, respectively: $Cr = \sum_{C \in \mathcal{C}} \delta^{int,C} \to max$ [LZXC14], $Cr = \sum_{C \in \mathcal{C}} \delta^{ext,C} \to min$.
  Another popular C-type criterion is conductance of a network partition: *Conductance* [YL12] of the partition (2.17) is maximum of conductances (2.13) of each community, and the goal is to decrease it:

$$\Phi = \max_{C \in \mathcal{C}} \phi(C) \to min. \tag{2.21}$$

  Finally, the most popular scoring function is modularity introduced by Newman [New04b]. It is based on comparison of partitions in the network and in the model-based random null model with the same degree distribution.

Modularity of the partition (2.17) in the unweighted network G is

$$M = \frac{1}{2m} \sum_{l=1}^{L^*} \sum_{u,v \in C_l} \left(a_{u,v} - \frac{d_u d_v}{2m} \right). \tag{2.22}$$

Here, the goal is increasing the scoring function (2.22).

- Second problem in CD is that even if a single scoring function is chosen, the problem of finding its optimum is NP-hard [Sch07],[BDG$^+$08].
- Third CD problem lies in a lack of reliable information about real communities, called *ground-truth communities* [YL12], for validation of CD results.

A variety of CDAs and their development is shown below with the help of popular algorithms implemented in R package IGraph:

1. **Girvan and Newman algorithm** (GNA, edge.betweenness.community in IGraph) [NG04] is a divisive hierarchical community detection algorithm, where division is based on removing edges with higher edge betweenness. The edge removal ends when modularity of resulting partition attains local maximum and can not be increased anymore by removing edges. Final communities are disconnected components of the partition. Complexity of the GNA is $O(n^3)$.
2. **Clauset** at al. **algorithm** (CA, fastgreedy.community in IGraph) [CNM04] is an agglomerative algorithm based on modularity optimization, which is much faster than a greedy technique proposed by Newman

[New04b]. It starts with the singleton clustering, where each node is a community, and then the procedure of adding edges is implemented, which provides increasing modularity in each iteration. This algorithm has complexity $O(n \cdot log^2 n)$ on sparse networks.

3. **Leading Eigenvector Algorithm** (leading.eigenvector.community in IGraph, LEA) by Newman [New06a] is a divisive spectral partitioning algorithm based on modularity optimization, which uses modularity matrix as a Laplacian. The generic algorithm divides a network into two communities depending on the result of rounding coordinates of the leading eigenvector to +1 or -1. It repeats iteratively for detecting more communities. Complexity is $O(n^2 log\, n)$ for sparse networks. Newman also proposed generalization of this basic method for arbitrary number of communities [New06a], [New06b].

4. Fast Modularity Optimization **Algorithm** by **Blondel** et al. (Louvain method, multilevel.community in IGraph, MLA) [BGLL08] is an agglomerative algorithm based on modularity optimization. On the first step modularity is maximized locally within neighbourhood of each node resulting in initial network partition into communities. On the second step it is optimized similarly on the weighted network with the communities as nodes. The process goes on until modularity increases. Complexity of the algorithm is unknown, but it is linear for typical and sparse data. This algorithm is good compromise between modularity

maximization and speed.

5. **Brandes** at al. Community Detection **Algorithm** (optimal.community in IGraph, OMA) [BDG+08]. It is also a greedy agglomerative CDA based on Linear Integer Programming application for modularity optimization. The iterative process starts with the singleton clustering and consequentially merges pairs of clusters yielding the largest increase or the smallest decrease of modularity, which is evaluated based on analysis of an increment matrix. Each iteration corresponds to moving to the adjacent vertex of a feasible polyhedron of an auxiliary linear program. Complexity is $O(n^2 log\, n)$.

6. **Spin Glass Algorithm** (spinglass.community in IGraph, SGA) [RB06], [TB09], [RN09]. It is a hierarchical agglomerative algorithm, which is based on the minimization of the Hamiltonian of the Potts-like spin model, where the spin states represent communities. "The community structure of the network is interpreted as the spin configuration that minimizes the energy of the spin glass with the spin states being the community indices" [RB06]. For a chosen resolution parameter and different initial conditions, partitions are found and their similarity is evaluated. Peaks of the similarity correspond to stable states of the system and, respectively, to the relevant partition. Complexity is slightly superlinear on the number of edges ($O(m^{1.3})$).

7. **Walk Trap Algorithm** by Pons and Latapy (walktrap.community in

IGraph, WTA) [YGFG+05] is a hierarchical agglomerative algorithm, which is based on random walks and starts from the singleton clustering. Then iteratively distances between current communities are computed and pairs of communities with the minimal value of a distance criterion are merged. WTA does not optimize modularity, but minimization of the distance criterion highly correlates with maximization of modularity. Complexity is $O(m^2 \cdot log\, n)$ for general cases and $O(n^2 \cdot log\, n)$ for sparse networks.

8. **Label Propagation Algorithm** by Raghavan, Albert and Kumara (LPA, RAK, label.propagation.community in IGraph) [RAK07] is an agglomerative algorithm, which uses, in addition to structural properties, propagation features of networks. It is iterative algorithm, where in each iteration every node sequentially updates its label to a new one with the most frequency among its neighbours (see Section 3.2.1). This algorithm is near linear on m on sparse data.

   The LPA is the fastest algorithm among all known and it allows adaptation for the attributed networks analysis (see Section 3.2.1 for details). Therefore we choose it and modify in Section 4.3.2.

It is important to note that different assumptions can arise regarding networks – such as acceptability of overlapping, existence of a hierarchy, restrictions on the number of communities and the amount of elements in them. Each of them leads to different approaches to CD (see [For10], [Sch07]).

2.4.1 Community Detection in Weighted Networks

Most of real networks are weighted [New04a]. They are considered as unweighted just for simplicity, whiles valuable information about the weights is lost. To avoid this problem and extract for CD as much as possible from available information, CDAs are adapted for weighted networks. For example, all mentioned above CDAs implemented in IGraph were at first designed for unweighted case. Currently they all are generalized for weighted networks as well. Main idea of the generalization [New04a],[HM14] is to consider unweighted multigraphs, instead of simple weighted graphs, with the number of multi-edges equal to the the corresponding edge weights in the initial weighted graph. As a result most of the algorithms are directly expanded to the weighted case. For example, modularity for weighted networks is obtained from (2.22) by replacement of: a) the adjacency matrix A by the weighted adjacency matrix A^w; b) the degree d_i of nodes by the node's strength s_i; c) the network's size m by its weight $\omega(G)$:

$$M^w = \frac{1}{2\omega(G)} \sum_{l=1}^{L^*} \sum_{u,v \in C_l} \left(a^w_{u,v} - \frac{s_u s_v}{2\omega(G)} \right) - \tag{2.23}$$

is modularity of the weighted network G^w. If G^w is normalised (see (2.3)), the formula (2.23) becomes $M^w = \frac{1}{2} \sum_{l=1}^{L^*} \sum_{u,v \in C_l} \left(w_{u,v} - \frac{1}{2} \cdot s_u s_v \right)$.

On the other hand, weighted networks have specific properties in comparison with unweighted ones, and far not every algorithm allows direct and easy extrapolation for the weighted case. One of such examples, when it is

possible to implement special techniques for the weighted case, is the LPA generalization described in Section 3.2.1. Also many CDAs designed directly for weighted networks were developed during last years (see a survey [HM14]), such as:

- "3 in 1" Center Based Algorithm [JPWX11]. It is an agglomerative algorithm including three stages, which repeat iteratively: a) finding centres of the networks - nodes with high degree and large weight; b) center adjusting for non-center nodes; c) community detection. Complexity of the algorithm is $O(n^2)$ and the number of communities is preassigned.

- a divisive Intra and Inter Centrality Algorithm (I^2C algorithm) [LWC13], which is based on conductance optimization [LLM10]. The iterative process starts with community $C \in \mathcal{C}$ of nodes connected by the highest weight edges. Then the expanding is carried out into an adjacent vertex $u \in V$ to C if: a) u has the highest belonging degree to C - $\frac{\omega(u,C)}{s_u}$; b) conductance (2.13) $\Phi(C \cup u) < \Phi(C)$. The expanding proceeds for the community C until conductance decreases, otherwise the community is detected. After that edges within the community are removed. The process is repeated until the edge set is empty.

2.5 Network Data Competion

Working with network data, we expect that the information is reliable, because only under this condition we can trust results of Network Analysis. Collecting reliable network data is one of the most challenging tasks [YL12]. Even nowadays, when extracting network data from web-networks is not a problem, it is still relevant for many other networks. For example, if we study a communication network of a certain group of people based only on their Internet contacts, it is not enough due to many other ways of communication. On the other hand, popular social web-network services accumulate some information about their users and form nodes' attribute data, which usually is not complete. It is because providing of this information is voluntary. Also pretty often information about users and their contacts is not complete or inaccurate due to the users, which intentionally hide or distort it. It is interesting to restore data in this case.

Network Topology Inference Problem (NTIP) [Kol09]. We deal with NTIP in case if the considered network is partially unobservable, and we want to infer this portion of missing data from measurements under known part of the network. There are different types of NTIP depending on available information and inference objectives:

1. *Link Prediction Problem* (LPP) [Kol09] is inferring whether or not a pair of vertices is connected by an edge based on measurements over observed set of edges.

2. *Network Completion Problem* (NCP) [KL11], where the issue is to complete missing parts of both nodes and edges.

3. *Network Attribute Inference Problem* (NAIP) is restoring attributes of edges, nodes, networks, and subnetworks.

Special cases of the above problems are: a) for LPP is Inference of Attributed Networks Problem, where new edges are established based on known edges and information about node attributes; b) for NCP is Inference of Tomographic Networks Problem [Kol09] - based on available information about nodes and edges located at the "perimeter" of networks, inferring edges and nodes in the "interior". There is also a special case of Inference of Attributed Networks Problem - Networks Inference of Association Networks [Kol09] - establishing whether or not a pair of vertices are adjacent based on the analysis of only node attributes.

In the NTIP class problem LPP was studied before [CMN08], [LKY+12]. NCP and NAIP are much less studied. One example of joint solution of NCP and NAIP is presented in [KL11]. First, there is proposed a solution of NAIP for recovering network's attributes and extracting the network type. Second, when NCP is solved, missing edges and nodes are recovered. One more approach to NAIP, related to restoring subnetworks attributes, is given in [THP08]. Authors propose a method using both edge and node attributes for forming a summation network, where node clusters play a role of new nodes. Then these enlarged nodes and edges are decorated by attributes

taken exceptionally from sets of node and edge attributes.

Chapter 3

Related Work

3.1 Association Network Inference Problem

If edges of an attributed network exist only between those vertices with sufficient level of their node attributes association, then such network is called an association network [Kol09].

Normally in a group of people we set up new connections easier with people that have hobbies and interests common to ours. But at the same time people with interests similar to ours not necessarily become our friends.

Comparing these occurrences in social networks, where for links existence it is necessary but not enough to reach some certain level of a node attributes association, in association networks an edge exists if an only if the level of association is attained.

If, based on ATNs, we know a function of the similarity $sim(i, j) = F(\overline{a}_i, \overline{a}_j)$,

where $\overline{a}_i, \overline{a}_j$ are vectors of attributes of nodes $v_i, v_j \in V$ (see (2.19)), then choosing a level of association α an edge set $E(G^a)$ of an association network G^a can be formed in a different way, e.g., according to the rule: $\{v_i, v_j\} \in E(G^a)$ if $F(\overline{a}_i, \overline{a}_j) \geq \alpha$.

Unfortunately, function $F(.)$ is usually unknown therefore typically the following assumptions are made that available network information is sufficient to restore missing elements of the network.

For example, [Kol09] focuses on a case, where the information, contained in node attributes, is sufficient to establish links. In particular, as a similarity function, there is recommended the correlation between vectors of node attributes $\overline{a}_i, \overline{a}_j$ - $sim(i,j) = \rho_{\overline{a}_i, \overline{a}_j}$ and is suggested to establish edges based on the results of verifying the following statistical hypothesises - $H_0^{ij} : \rho_{\overline{a}_i, \overline{a}_j} = 0$ versus $H_1^{ij} : \rho_{\overline{a}_i, \overline{a}_j} \neq 0$. Since links in association networks are formed based only on a similarity of node attributes, the described algorithm is a way to solve Networks Inference of Association Networks Problem (see Section 2.5).

A drawback of the technique [Kol09] is that the correlation does not count weights of node attributes. Moreover, this approach does not work if only one node attribute is present.

The first disadvantage can be overcomed by using "weighted" generalization of the standard correlation.

3.2 Relevant Community-Detection Algorithms

3.2.1 Label Propagation Algorithm and Modifications

Since networks can have tens or even hundreds of millions of nodes, the highest priority for CDAs is their computational efficiency that can guarantee required level of CD quality.

The LPA has nearly linear complexity [RAK07], besides it is easily parallelized and distributed. The algorithm was proposed by Raghavan in 2007 [RAK07]. Since that time it became very popular and has been modified and improved in several ways [BC09],[Gre10],[LM10],[KPS13]. Essentially, the LPA uses propagation features of networks, meaning that nodes adopt new characteristics depending on the behaviour of their neighbours, e.g. adopts labels of the biggest amount of its neighbours.

Briefly the LPA works as follows:

- Step 1. An initial iteration consists of assigning unique labels to each node;
- Step 2. A random order of nodes' revision is established and is used on Steps 3-4 iteratively;
- Step 3. Each node is revised in the assigned order and adopts the most frequent label of its neighbours: for $x \in V$

$$l_x^{new} = \underset{l}{argmax} \sum_{u \in V} a_{ux} \delta(l_u, l), \tag{3.1}$$

where l_x, l_x^{new} are current and new labels of a node x, respectively, $(a_{ux})_{u,x\in V}$ is an adjacency matrix, $\delta_{l_u,l} = \begin{cases} 1, \text{ if } l_u = l; \\ 0, \text{ otherwise .} \end{cases}$

Breaking Ties Rule: ties are broken randomly implying that if the label choice (3.1) is not unique, then the following random label selection is applied: $l_x^{new} = sample(\underset{l}{Argmax}(x), 1)$ - a random sample of size 1 from the set of these candidates:

$\underset{l}{Argmax}(x) = \{l^{max} : \sum_{u\in V} a_{ux}\delta(l_u, l^{max}) = \underset{l}{max} \sum_{u\in V} a_{ux}\delta(l_u, l)\}$.

- Step 4. The process is performed iteratively until algorithm converges and no label changes occur anymore.

The main drawback of the LPA is its instability and sensitivity to the order of labels update (see Steps 2-3) [BC09]. To avoid this disadvantage it was suggested to use a directed search of extremum of the scoring function, such as modularity (2.22). For example, Barber and Clark in [BC09] introduced a modularity-specialized LPA (LPAm), where instead of the new label choice rule (3.1) the following one is used:

$$l_x^{new'} = \underset{l}{argmax} \sum_{u\in V} b_{ux}\delta(l_u, l), \text{ where } b_{uv} = a_{uv} - \frac{d_u d_v}{2m}. \tag{3.2}$$

This rule (3.2) guarantees modularity increasing. Modularity can have many local maxima. So to avoid getting stuck in poor ones, Liu and Murata [LM10] improved the LPAm in an advanced modularity-specialized label propagation algorithm (LPAm+). In the LPAm+ a gain of modularity is obtained in

the following way: if M improvement is not possible by adopting labels by a single vertex, then in order to escape the local maximum of M pairs of communities are matched.

All these methods are generalised for weighted graphs: a) for the LPA the new label choice rule (3.1) becomes:

$$l_x^{new} = \underset{l}{argmax} \sum_{u \in V} a_{ux}^w \cdot \delta(l_u, l), \tag{3.3}$$

b) for the LPAm and the LPAm+, which are modularity based, the rule (3.2) becomes:

$l_x^{new} = \underset{l}{argmax} \sum_{u \in V} b_{ux}^w \delta(l_u, l)$, where $b_{uv}^w = a_{uv}^w - \frac{s_u s_v}{2\omega(G)}$.

The rule (3.3) is implemented in the IGraph LPA and it can be interpreted in the following way: at Step 3 each node adopts a label of its neighbours with the biggest total edge weight; if the choice is not unique, then new label is chosen from candidate labels randomly.

One more popular modification of the LPA is a Community Overlap Propagation Algorithm (COPRA), described by Gregory [Gre10]. It works as with weighted networks and as with overlapping communities and has complexity $O(m \cdot\ n)$. It uses probabilities of vertices to belong to a given community, belonging coefficient of vertices to communities, it keeps labels with belonging coefficients exceeded the threshold $\frac{1}{L^{max}}$, where L^{max} is the maximum number of communities. If the choice is not unique it applies the LPA Breaking Ties Rule and then normalises the remaining belonging coefficients. If $L^{max} = 1$

then the COPRA becomes the LPA.

3.2.2 Label Propagation Algorithms with Seeds

Seed based algorithms, which deal with partially known communities, represent prospective direction of CD and consider many relevant issues in the area [KK14], [LZXC14], [SMM14].

One of them is *Seed Set Expansion Problem* (SSEP) [KK14]. Assume that we have a graph G that contains a group of nodes $C \subseteq V(G)$, which identities need to be uncovered using knowledge about identities of its subset $S \subset C$. Subset S is called a *seed set* or a *seed*. If C is community ($C \in \mathcal{C}$) then the SSEP is the problem of recovering the whole community C from its seed S. The SSEP has many applications [KK14] and can be generalised for several communities. Depending on acceptability of overlapping, we come to the following two problems of G-partition or G-cover recovering:

- *Seed Sets Expansion Partition Problem* (SEPP): if $S_l \subset C_l$, $l \in J_L$, are C_l-seeds in the partition (2.5) and the task is to find the partition $\Pi(G)$;
- *Seed Sets Expansion Coverage Problem* (SECP): if $S_l \subset C_l$, $l \in J_L$, are C_l-seeds in a cover $\Pi(G)$ (see (2.5)) and this cover should be found.

The SSEP is a subproblem of a *Selective Community Detection Problem* (SCDP) [SMM14], which is solved in two steps. The problem of seeds selection, which is in fact SSEP, is solved on the first stage. On the second stage the seeds are expanded. SCDP has many advantages due to a number of applications

and, at the same time, operates with a small amount of information. In particular, for the first stage performing only a network structure is needed. When seeds are already chosen, then for the whole community recovering it is sufficient to get exogenous information only about the real communities, which these seeds belong to. The remaining part of communities is obtained at the second stage execution.

Normally single nodes are considered as seeds [LZXC14],[SMM14],[YL12] and called *seed nodes*. This partial type of SSEP is called a *Seed Nodes Expansion Problem* (SNEP). Generally Seed Sets Expansion CDAs are based on random walkings from seeds [SMM14].

One more prospective direction in seed sets expansion algorithms is adaptation of the LPA [LZXC14], for instance, exactly this version of the LPA currently is implemented in IGraph. The LPA application seems natural for seed sets expansion because the LPA can start from any initial label distribution and then, since seeds labels remain unchanged, it needs a slight modification for updating just non-seed nodes (see Section 4.3.2). For instance, Lin et al. [LZXC14] are presented a Community-Kernel LPA (CK-LPA). It is stable and hence it converges due to utilization of weights of nodes. It consists of three consecutive steps: a) choose a set of disconnected nodes of the maximum degree and set it as a seed; b) detect a network kernel that is a partition into clusters with high total density; c) assign unique label to each of the clusters and propagate the label from the initial state.

3.2.3 Community Detection in Attributed Networks

Standard CDAs [For10] at most work with weighted graphs, thus, they use fully only a topological structure of networks, partially - edge attributes accumulated in edge weights, and totally ignore node attribute information. At the same time information about both, edges and nodes, is important. For instance, in social networks edges describe relationships and their weights strongly depend on roles of participants, which can be represented as node attributes. Algorithms of CD specifically for attributed networks (ANCDAs) are developed for utilizing available in (2.14) information. Presence of different node attributes means heterogeneity of vertices, also different edge attributes imply that edges are heterogeneous too. An ideal CDA for attributed networks should provide balance between structural and node attributes commonalities and should generate dense clusters with homogeneous vertices' and edges' properties. It is quite challenging task, because these three goals - dense connection, edges and nodes homogeneity - can conflict.

Let us briefly review several approaches to CD for attributed networks. In [THP08] Tian et al. propose graph summarization approach that generate clusters, primarily based on similarity of node attributes, and, at the same time, count edge attributes. Methods introduced in [THP08] and [BHS13] represent a group of ANCDAs, which are based on *user-selected node attributes* and combine graph clustering with subspace clustering, where subspace is defined by the selected attributes. Zhou et al. present a SA-Cluster Algorithm [ZCY09] and then improve it in an Inc-Cluster Algorithm [ZCY10], where new

edges are added based on nodes similarity, vertices with identical attributes are connected through additional vertices of node attributes and the Random Walk CDA (see Section 2.4) is applied for constructing augmented graph, where random walk distance matrix is effectively computed by matrix increments. SA-Cluster and Inc-Cluster algorithms are *distance-based* ANCDAs. For such algorithm class an artificial distance measure combining node attributes and structural and information is designed. It uses weights W^I, W'^I (see Section 4.3) of both these parts, respectively. A drawback of such type of algorithms is that the result of CD highly depends on the parameters W^I, W'^I, which can not be extracted directly from networks, so, an estimation of this exogenous data might be costly. Another way of the attributed networks analysis is a *model-based* approach, where a null-model is designed for node and edge information consideration. For instance, Xu et al. in [XKW+12] propose an Bayesian Attributed Graph Clustering (BAGC) algorithm, where the following assumptions are used for the null model: a) the true partition exists but but it is unknown; b) vertices from the same community behave similarly whiles nodes from different communities may behave differently. Bayesian model was used for defining a joint probability distribution, which transforms the attributed network CD problem into a standard probabilistic inference problem that can be solved by specific variational algorithm. Yang et al. [YML13] proposed another [YML13] model-based method CESNA (Communities from Edge Structure and Node Attributes) that in addition to the assumptions a) and b) includes next ones: c) nodes from the same

communities most likely are adjacent; d) nodes may belong to multiple communities; e) if more than two nodes belong to the same community, then most likely they are adjacent.

Two of the authors [YML13], Yang and Leskovec, devoted other research works to analysis of attributed networks, especially to ANCDAs [YL12],[YL13]. These researches validate their algorithms on a number if real networks such as LiveJournal, Friendster, Orkut, Amazon, DBLP with explicit participants characteristics. For instance, in the popular social network LiveJournal it is divisions according to culture, entertainment, life/style, gaming, sports, technology, etc. The real groups of people are considered as ground-truth communities (GTCs), which are used for validation of different hypothesis and results of CD [YL12],[YML13],[YL13]. It turned out [YL12] that the GTCs are very different from standard "structural" communities, since CDAs attempt to find tightly connected groups of nodes, which are "structural" communities, whiles the real GTCs are well separated from each other and not necessary well connected inside. A comparison of the sensitivity of numerous scoring functions and how they impact a given community detection algorithm is given [YL12] along with a CDA based on a local spectral clustering, applying different community scoring functions, and solving SNEP (see Section 3.2.2) from one seed in each ground-truth community. The results confirmed the hypothesis that for CD of the GTCs is better to use measures of separability such as conductance. The authors also state that CDAs for GTCs should allow overlapping the communities since the network participants belong

to various number of GTCs of different categories that typically overlap. The mentioned local spectral clustering algorithm [YL12] allows to detect overlapping communities by choice of seed nodes in different GTCs. The works [YML13], [YL13] continue the developing overlapping ANCDAs, but the used approach is entirely different - it is a model-based one. For instance in [YML13] there is proposed a graph model that is able to generate networks with community structure entirely based on the probability of pairs of nodes affiliation to GTCs. The probability serves by a similarity function and is called affiliation function, and the described above CDA CESNA is based on it. The researchers continued developing the direction in [YL13] and presented a CDA based on a Cluster Affiliation Model for Big Networks (BIGCLAM). The method uses the same affiliation function and tries to fit nodes to their most likely attribute affiliations according to a model of maximum likelihood, when node attribute assignment is approached as an optimization problem.

3.3 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) is a multi-objective multi-criteria decision making method, invented by Saaty [Saa77],[SA89]. An idea of the method is to compare pairwise criteria in order to obtain relative weights of elements of choice.

The challenging part of this method is in assigning weights (global priorities) of alternatives. If alternatives are compared with respect to criteria of the next

top level, then these criteria, in turn, are evaluated depending on criteria of the next top level and so on. Weights of all criteria against the next top level ones form vectors of local priorities, which can be computed accurately for numerical criteria functions or can be assessed based on leading eigenvectors of preference matrices. Prioritisation of decisions is made at the stage of synthesis of local priorities vectors into a global priorities vector, which dimension coincides with ones of local priorities vectors of the decision alternatives. The global priorities vector is a linear combination of these vectors with coefficients depending on local priorities of all criteria and alternatives. Normalization of the local priorities vectors guarantees normalization of the resulting global priorities vector.

Chapter 4

Attributed Networks Analysis

This chapter is dedicated to the analysis of attributed networks: a) in Section 4.1 there is given a classification of attributed networks; b) in Section 4.3.1 all available information about any attributed network is utilized in an *aggregated network*; c) Section 4.3.2 is dedicated to attributed network applications, namely for Problems 1-3 introduced in Chapter 1; d) Problem 4 is considered in Section 4.4 (see Chapter 1).

There are different approaches for dealing with attributed networks (see Chapter 3). The one, developed by us, consists of the following - combine all node attributes of the original network G in an association network G^a, then aggregate all edge attributes in a weighted network G^w, lastly, combine both these networks in an aggregated network G^{wa} and study its structure, particularly its community structure.

4.1 Classification of Attributed Networks

Recall that node and edge attribute values are given by $n \times K$, $m \times K'$ matrices Λ, Λ', respectively (see (2.20)). Let us suppose that some attribute values can be missing. To reflect the situation we represent such values by 0-entries of the matrices and add 0-values to the set of attributes values $\{at^{nk}, at^{ek'}\}_{k \in J_K, k' \in J_{K'}}$. We assume that elements $a_i^k, a_{\{i,j\}}^{'k} \neq 0$ in tuples (2.19) correspond to the available attributes and $a_i^k, a_{\{i,j\}}^{'k} = 0$ - to the missing ones $(i, j \in J_n)$.

Now the form (2.14) can represent not only social networks, but also any unweighted or weighted networks, unattributed or attributed networks, and among the last ones - a CIAN or an IIAN:

- unattributed networks:
  - if $\Lambda = \Lambda' = \emptyset$ then the network G is an unweighted network $G = (V, E)$;
  - if $\Lambda = \emptyset, \Lambda' = \lambda' \in R^m$ then the network G is weighted network $(G = (V, E, \lambda'))$ with weights given by the numerical vector λ';
- attributed networks: a) $K + K' > 0$; b) if $K = 0, K' = 1$ then $\Lambda' = \lambda' \notin R^m$ (for instance, λ' is ranking).

  Depending on combinations of K, K' attributed networks can be:
  - just *node attributed* unweighted *networks*, if $K > 0, K' = 0$, $G = (V, E, \Lambda)$, or node attributed weighted networks if $K > 0, K' = 1$,

$\lambda' \in R^m$, $G = (V, E, \Lambda, \lambda')$;

- just edge attributed networks if $K = 0, K' > 0$, $\Lambda' \notin R^m$, $G = (V, E, \Lambda')$;
- node and edge attributed networks if $K, K' > 0$ (G has form (2.14));
- *multi-attributed networks* if $max(K, K') > 1$.

It should be also noted that attributed networks can be classified depend on availability node-edge attribute values:

- a CIAN is characterised by non-zero values of all node and edge attributes, which can be assumed positive: if $K > 0 \Rightarrow \Lambda > 0$; if $K' > 0 \Rightarrow \Lambda' > 0$;
- an IIAN, where some attributes are missing and therefore exist at least one zero values of Λ, Λ': $\Lambda' \not> 0$ or $\Lambda \not> 0$.

Also we can classify attributed networks depending on types of available attributes. The type of attributed networks depends on types of their node and edge attributes. The last ones can be *numerated* (*N-type*) or *enumerated* (*E-type*). In turn, N-type attributes can be *discrete* (*D-type*) or *continuous* (*C-type*), and E-type attributes can be *ranking* (*R-type*) or *unranked* (*U-type*). The type of an attributed network is defined by tuples of single attributes types $\overline{AT}^n_{type} = \{AT^{nk}_{type}\}_k, \overline{AT}^e_{type} = \{AT^{ek'}_{type}\}_{k'}$, specifically by sets of different attribute types $S(\overline{AT}^n_{type})$ and $S(\overline{AT}^e_{type})$ presented in the network,

where $\{AT^{nk}_{type},\ AT^{ek'}_{type}\} \in \{D, C, R, U\}\ \forall k, k'$. For attributed networks the following notation $AN(S(\overline{AT}^{n}_{type}), S(\overline{AT}^{e}_{type}))$ is used. For instance, an usual unattributed network will be classified as $AN(\emptyset, D)$ since its edges can be considered as ones of the weight 1. $AN(\emptyset, N)$-class includes weighted networks which have the only numerical edge attribute "weight" ($K' = 1$). It is obvious that introducing attributed networks as a generalisation of usual unweighted and weighted networks we suppose presence at least one of the following: a) an ATN, b) a number of ATEs, c) E-type edge attributes. It is also clear that for dealing with several different attributes or with ones of different types we need special techniques. So, this is the right point to rise several questions, which will be answered in the future sections (see Section 4.3):

- how to combine ATNs or ATEs of different types?
- how to convert E-type into N-type?
- how to build auxiliary networks of single node attributes and then merge them into an association network?
- how to construct auxiliary networks of single edge attributes and then combine them into a weighted network?

Let us note main advantages and drawbacks of the listed attribute types:

- R-type ATEs are more preferable than U-type ones, because their ordering allows to transform them into numerical values faster since

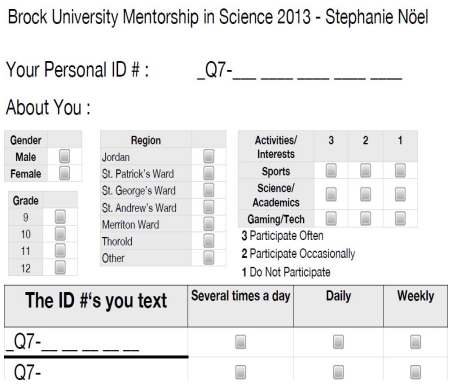
a feasible region of these parameters is smaller (see an example in Section 5.4.2);

- C-type attributes contain more information than of D-type, because the last one is normally used for simplification of reality. At the same time, for attributed networks analysis usage of C-type node attribute values is not preferable, because, even though they allow a numerical estimation of nodes similarity (see Section 3.1), they convert the attributed network into a near complete graph, that makes the approach inapplicable for large-scale networks. At the same time, discrete node attribute values lead to overlapping of partitions by complete graphs (see Section 4.3.1 and an example in Section 5.1).

4.2 The HSTN description

In order to construct G 398 high-school students of Denis Morris Catholic High School (Thorold, Ontario) were asked to provide information about their texting contacts, gender ("Gn"), grades ("Gr"), residence location (Region, "R") and attitude to activities (see Figure 4.1). Intensity of: a) texting contacts ("TC"); b) participation in the following activities: Sports ("S"), Science/Academics ("Sc"), and Gaming/Tech ("Ga") is represented by three categories and is ranked from worst to best. For instance, "1" corresponds to the weakest case ("cold" contacts and activities participation), "3" - to the strongest one (a "hot" contact and not participation in an activity), and "2" -

to medium one ("worm" contacts and rare participation in the corresponding activity). The survey was conducted by a high-school student Stephanie Noel. From each of the completed questionnaires a directed star graph was formed (see Figure 4.2) and then all these graphs were combined into one undirected graph. In case if two participants included each other in their texting contact lists, then such intensity of the mutual contact was established as the highest one.


Brock University Mentorship in Science 2013 - Stephanie Nöel

Your Personal ID # : _Q7-__ ___ ___ ___ ___

About You :

Gender
Male
Female

Grade
9
10
11
12

Region
Jordan
St. Patrick's Ward
St. George's Ward
St. Andrew's Ward
Merriton Ward
Thorold
Other

Activities/ Interests	3	2	1
Sports			
Science/ Academics			
Gaming/Tech			

3 Participate Often
2 Participate Occasionally
1 Do Not Participate

The ID #'s you text	Several times a day	Daily	Weekly
_Q7-__ __ __ __ __			
Q7-			

Figure 4.1: The questionnaire

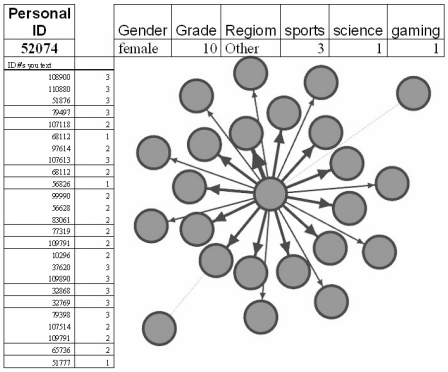

Personal ID		Gender	Grade	Regiom	sports	science	gaming
52074		female	10	Other	3	1	1
ID #'s you text							
108900	3						
110880	3						
51876	3						
76407	3						
107118	2						
68112	1						
97614	2						
107613	3						
68112	2						
56826	1						
99990	2						
56628	2						
83061	2						
77319	2						
109791	2						
10296	2						
31620	3						
106990	3						
32868	3						
32769	3						
79398	3						
107514	2						
109791	2						
65736	2						
59777	1						

Figure 4.2: The extracted part of the HSTN from the completed questionnaire

As a result the High School Texting Network (HSTN) with $n = 521$ nodes and $m = 1887$ edges was built. The network is decorated by node attributes "Gn","Gr","RL","S","Sc","Ga" and by an edge attribute "TC". It is an IIAN because from one standpoint a part of participants did not provide complete information, and from other standpoint $521 - 398 = 123$ students were mentioned in the contact lists but did not participate in the survey. For these 123 students all the information, except for a partial list of

texting contacts and their intensity, is missing.

The HSTN has the following parameters (see Tables 4.1,4.2): a) there are K node attributes ($K = 6$) and K' edge attribute ($K' = 1$); b) node and edge attributes are $\overline{AT}^n = (AT^{nk})_{k\in J_6} = (Gn, Gr, RL, S, Sc, Ga)$ and $\overline{AT}^e = (AT^{e1}) = (TC)$, respectively; c) the node attributes take $\{L_k\}_k$ values ($\{L_k\}_k = \{2, 3, 3, 3, 7, 4\}$) from $\overline{at}^n = (at^{nk})_{k\in J_6} = (\{female, male\}, \{1, 2, 3\}, \{1, 2, 3\}, \{1, 2, 3\}, \{1, 2, 3\}, \{Jordan, ..., Thorold\}, \{9, 10, 11, 12\})$; d) the edge attribute takes 3 values ($L'_1 = 3$) from $\overline{at}^e = at^{e1} = \{1, 2, 3\}$.

A robust subnetwork, which is a CIAN G' on $n' = 348$ nodes and $m' = 1148$ edges, included those the survey participants who provided all the data, was extracted from G. Table 4.1 represents $\{n_l^k, n_l'^k\}_{l,k}$ - frequencies of ATNVs in G and G', respectively. Table 4.2 shows $\{m_l^1, m_l'^1\}_l$ - frequencies of ATNEs values in G and G', respectively.

The percentage of missing node attribute values in G is 29.2%, and the percentage of incompleteness data in G' is $\frac{n-n'}{n} = 33.2\%$.

The HSTN is categorised according to the classification presented in Section 4.1. The node attribute types are $\overline{AT}^n_{type} = \{\overline{AT}^{nk}_{type}\}_{k\in J_6} =$ $\{U, R, U, R, R, R\}$. Respectively, the HSTN type is $AN(S(\overline{AT}^n_{type}), S(\overline{AT}^e_{type})) =$ $AN(S(U, R, U, R, R, R), S(R)) = AN(\{U, R\}, R)$.

Notice that the only texting intensity ranks can be used because "TC" values are ordered in the same way, from worst ("cold") to best ("hot") contacts, for the whole contact network. So, we assume that in our case intensive texting implies close communication and, hence, leads to communities formation. The

l	Gender	n_l^1	$n_l^{'1}$	Sports	n_l^2	$n_l^{'2}$	Science	n_l^3	$n_l^{'3}$	Gaming	n_l^4	$n_l^{'4}$
1	female	202	189	1	88	77	1	123	109	1	169	153
2	male	167	159	2	84	77	2	137	128	2	101	98
3	0	152		3	201	194	3	112	111	3	102	97
4				0	148		0	149		0	149	
Total		521	348		521	348		521	348		521	348

l	Region	n_l^5	$n_l^{'5}$	Grade	n_l^6	$n_l^{'6}$
1	Jordan	18	17	9	93	90
2	Merriton Ward	61	60	10	78	74
3	Other	150	145	11	109	103
4	St. Andrew's Ward	5	5	12	85	81
5	St. George's Ward	4	4	0	156	
6	St. Patrick's Ward	1	1			
7	Thorold	123	116			
8	0	159				
Total		521	348		521	348

Table 4.1: Frequencies of the node attribute values in the HSTN (n_l^k in G, $n_l^{'k}$ in G')

at_l^e	m_l^1	$m_l^{'1}$
1	760	431
2	625	387
3	502	330
Total	1887	1148

Table 4.2: Frequencies of the edge attribute values in the HSTN (m_l^1 in G, $m_l^{'1}$ in G')

ranks are also assigned to the ATNs "Gr","S","Sc","Ga", but they are local and are related just to the grade and particular activities. Thus, these ranks hardly can be applied to the whole network. For example we do not have any information about importance for the HSTN community structure of 9-th grade high-school students in comparison to 10-th grade ones. The same is true for the rest ranking ATNs - "S","Sc","Ga". Obviously, the latter are

related to high-school students attitude to the activities and do not explain anything about closeness of the contacts. For example, we can think that frequent participation in sports implies close relationships within a sport community, whiles those who do not do sports have less communication. At the same time a real situation can be opposite - a student concentrated on sport may have not enough time for communication, whereas a student who do not do any sports may have a lot of time for contacting with friends. For clarifying these issues an additional analysis, such as an expert assessment, is needed.

Remark 1. *Even if the HSTN G can be classified as $AN(\{U, R\}, R)$, due to lack of this information, in the scope of the thesis we consider it as $AN(U, R)$. There are several transformations of G to be done before applying CD: 1) at first we convert the edge ranked part G into numerical weights, in such a way G is converted into a weighted network G^w of type $AN(\emptyset, N)$ (see Section 5.4.2); 2) then it's node ranked part is converted into weights of an association network G^a of the same type in Section 5.4.4; 3) and finally, all the available structural and attributed network information accumulated in G^w, G^a, respectively, is combined into an attributed network G^{wa}, which is again usual weighted network of the type $AN(\emptyset, N)$ (see Section 5.4.4). Similar transformations is performed for the robust HSTN G' in Section 5.4.3.*

4.3 Accumulation of Network Information and its Application

The scheme that is used for the HSTN analysis (see Remark 1) can be also applied to any attributed network. For forming the attributed network G^{wa} we use additional information of three levels: a) weights of the association network G^a and weighted network G^w in G^{wa}-formation (Level I); b) weights of individual ATNs and ATEs (Level II); c) weights of single ATNVs and ATEVs (Level III).

We form two sets of auxiliary networks: a) $\{G^{ak}\}_{k\in J_K}$ - a set of association networks corresponding node attributes $\{AT^{nk}\}_{k\in J_K}$; b) $\{G^{wk}\}_{k\in J_{K'}}$ - a set of weighted networks corresponding edge attributes $\{AT^{ek}\}_{k\in J_{K'}}$.

Introduce some notations:

- Level I - $\overline{W}^I = (W^I, W'^I)$ is a vector of weights of G^a and G^w, respectively;
- Level II - a) $\overline{W}^{II} = (W^{II,k})_{k\in J_K}$ is a tuple of ATNs-weights in G^a; b) $\overline{W'}^{II} = (W'^{II,k})_{k\in J_{K'}}$ is a tuple of ATEs-weights in G^w;
- Level III - a) $\overline{W}^{III,k} = (W^{III,lk})_{l\in J_{L_k}}$ is a tuple of weights of ATNVs in G^{ak} $(k \in J_K)$; b) $\overline{W'}^{III,k} = (W'^{III,lk})_{l\in J_{L'_k}}$ is a tuple of weights of ATEVs in G^{wk} $(k \in J_{K'})$;
- all these tuples are positive: $\overline{W}^I$, $\overline{W}^{II}$, $\overline{W'}^{II}$, $\overline{W}^{III}$, $\overline{W'}^{III} > 0$ (otherwise dimension of the problem can be reduced) and normalized:

$$||\overline{W}^{I}|| = ||\overline{W}^{II}|| = ||\overline{W'}^{II}|| = ||\overline{W}^{III,k}||_{k\in J_K} = ||\overline{W'}^{III,k'}||_{k'\in J_{K'}} = 1. \quad (4.1)$$

The weights W^I, W'^I can be interpreted as priorities of the networks G^a, G^w in the G^{wa}-network structure. Similarly, $\overline{W}^{II}$, $\overline{W'}^{II}$ are priorities of the auxiliary networks $\{G^{ak}\}_{k\in J_K}$, $\{G^{wk}\}_{k\in J_{K'}}$. Finally, $\{\overline{W}^{III,k}\}_k$, $\{\overline{W'}^{III,k'}\}_{k'}$ are priorities of subnetworks corresponding to single attribute values.

The weights can be obtained in different ways, for instance, by an expert assessment or derived directly from the network (see Section 4.3.1). An example of the analysis and extracting $\overline{W}^{II}$ and $\overline{W}'^{III,k}$ directly from the HSTN is described in Section 5.4.2.

Remark 2. *If information about some of Levels I-III is not available, then we suppose that the corresponding weights in* (4.1) *are equal.*

4.3.1 Attributed Network Construction

Node Information Utilization

The One Discrete Node Attribute Network G^{ak} Formation We build the association network G^a as a weighted network sum of the auxiliary networks $\overline{G}^a = \{G^{ak}\}_{k\in J_K}$ of the single node attributes $\{AT^{nk}\}_{k\in J_K}$.

Introduce *node attribute clusters* (ACs) and G-partition into them. Let

$$\mathcal{AC}^k = \{AC_l^k\}_{l\in J_{L_k}} \quad (4.2)$$

denote a G-partition into ACs related to each value of AT^{nk} and

$$\overline{\mathcal{AC}} = (\mathcal{AC}^k)_{k \in J_K} \tag{4.3}$$

denote a tuple of partitions into these clusters. By (2.19) each node attribute cluster (AC) with at_l^{nk}-value of AT^{nk} is represented as follows:

$$AC_l^k = \{v_i \in V : a_i^k = at_l^{nk}\},\ |AC_l^k| = n_l^k,\ \sum_{l \in J_{L_k}} n_l^k = n\,(l \in J_{L_k},\ k \in J_K), \tag{4.4}$$

Let us describe an order of constructing the network $G^{ak} \in \overline{G}^a$. It should be normalized (Condition 1), should have edges between nodes with the same value of AT^{nk} (Condition 2), edge weights within different attribute clusters AC_l^k and $AC_{l'}^k$ $(l \neq l')$ should be proportional to the priorities of the values at_l^{nk}, $at_{l'}^{nk}$ in G^{ak} (Condition 3), and finally, it should have equal weights within the same attribute cluster (Condition 4).

In our notations these conditions can be represented as follows:

- Condition 1 - the normalization (see (2.3)):

$$\omega(G^{ak}) = 1; \tag{4.5}$$

- Condition 2 - the edge set formation: $\forall i, j \in J_n$ an edge $\{v_i, v_j\}$ exists iff

$$\exists l \in J_{L_k} : a_i^k = a_j^k = at_l^{nk}; \tag{4.6}$$

- Condition 3 - the edge weights distribution: if (4.6) holds and

$$\exists l' \in J_{L_k} : a^k_{i'} = a^k_{j'} = at^{nk}_{l'} \Rightarrow \tag{4.7}$$

$$\frac{w^{ak}_{ij}}{w^{ak}_{i'j'}} = \frac{W^{III,lk}}{W^{III,l'k}}, \tag{4.8}$$

  where $\overline{w}^{ak} = (w^{ak}_{ij})_{i,j\in J_n}$ is a WAM of G^{ak};

- Condition 4 - equal weights within AC^k_l:

$$\text{if for } i,i',j,j' \in J_n \text{ (4.6), (4.7) hold and } l = l' \Rightarrow w^{ak}_{ij} = w^{ak}_{i'j'}. \tag{4.9}$$

Notice that (4.6) can be rewritten as following:

$$\exists l \in J_{L_k} : v_i, v_j \in AC^k_l. \tag{4.10}$$

To satisfy (4.8), (4.9) we choose weights $\{w^{ak}_{ij}\}_{i,j}$ proportionally to $W^{III,lk}$:

$$w^{ak}_{ij} = \nu(k) \cdot W^{III,lk} \text{ if (4.10) holds, otherwise } 0\,(i,j \in J_n), \tag{4.11}$$

where $\nu(k)$ is a normalized factor of G^{ak}.

Define $\nu(k)$ using (4.4), (4.5), (4.11), and the fact that edges exist only between nodes with the same ATNV implying a partition of the network by a disjoint union of complete graphs:

$1 = \omega(G^{ak}) = \sum_{i,j} w^{ak}_{ij} = \sum_{l\in J_{L_k}} \sum_{i,j\in AC^k_l} w^{ak}_{ij} = \sum_{l\in J_{L_k}} \sum_{i,j\in AC^k_l} \nu(k) \cdot$

$W^{III,lk} = \nu(k) \cdot \sum_{l \in J_{L_k}} W^{III,lk} \cdot 2||K_{n_l^k}|| = \nu(k) \cdot \sum_{l \in J_{L_k}} W^{III,lk} \cdot n_l^k \cdot (n_l^k - 1)$

wherefrom

$$\nu(k) = \left(\sum_{l \in J_{L_k}} W^{III,lk} \cdot n_l^k \cdot (n_l^k - 1) \right)^{-1}, \; k \in J_K. \tag{4.12}$$

Summarise the result in the following observation:

Observation 1. *Each association network G^{ak} satisfied* (4.5)-(4.9) *is a G^a-partition by complete graphs $\{K_{n_l^k}\}_{l \in J_{L_k}}$ with the WAM $\overline{w}^{ak}$ defined by* (4.11) *and $\nu(k)$ defined by* (4.12) *($k \in J_K$). We can represent it as follows:*

$$G^{ak} = G^a[\mathcal{AC}^k] = \underset{l \in J_{L_k}}{\cup} G^a[AC_l^k] = \underset{l \in J_{L_k}}{\cup} K_{n_l^k}, \; k \in J_K. \tag{4.13}$$

Remark 3. *Notice that $\overline{W}^{IIIk}$ can be not normalized, but weights* (4.11) *do not change if multiply $\overline{W}^{IIIk}$ by non-zero factor (see* (4.12)*). Therefore, without loss of generality, assume that for $\{\overline{W}^{IIIk}\}_{k \in J_K}$* (4.1) *holds.*

Remark 4. *If $\overline{W}^{IIIk}$ is unknown, then: a) by Remark 2 weights are equal ($W^{III,lk} = W^{III,l'k}$, $\forall l, l' \in J_{L_k}$); b) by Remark 3 they are normalized, hence, $W^{III,lk} = \frac{1}{|\overline{W}^{III,k}|} = \frac{1}{L_k}$ and the formulas* (4.12) *and* (4.11) *become $\nu(k) = \frac{L_k}{\sum_{l \in J_{L_k}} n_l^k \cdot (n_l^k - 1)}$, $k \in J_{L_k}$,*

$$w_{ij}^{ak} = \left(\sum_{l \in J_{L_k}} n_l^k \cdot (n_l^k - 1) \right)^{-1} \text{ if } (4.10) \text{ holds, otherwise } 0 \, (i, j \in J_n). \tag{4.14}$$

The Discrete Association Network G^a Construction The association network G^a is formed as a weighted network sum of networks $\overline{G}^a$ with the weights $\overline{W}^{II}$:

$$G^a = \sum_{k \in J_K} W^{II,k} \cdot G^{ak}. \tag{4.15}$$

Lemma 1. *If* (4.5) *holds then* G^a *is normalized:*

$$\omega(G^a) = 1. \tag{4.16}$$

Proof. *Using* (4.1), (4.5), (4.15) *we have* $\omega(G^a) = \omega(\sum_{k \in J_K} W^{II,k} \cdot G^{ak}) = \sum_{k \in J_K} W^{II,k} \cdot \omega(G^{ak}) = \sum_{k \in J_K} W^{II,k} = 1$. *So, the* G^a*-normalization condition* (4.16) *holds.*

Notice that a vertex set of G^a is the same as for the original network: $V(G^a) = V$, its edge set $E(G^a)$ is a union of $\{G^{ak}\}_k$ edges sets: $E(G^a) = \cup_{k=1}^{K} E(G^{ak})$, the WAM $\overline{w}^a = (w^a_{ij})_{i,j}$ is the following linear combination of G^{ak}-WAMs: $\overline{w}^a = \sum_{k \in J_K} W^{II,k} \cdot \overline{w}^{ak}$ ($\overline{w}^{ak} = (w^{ak}_{ij})_{i,j}$, $k \in J_K$).
If the node attribute weights $\overline{W}^{II}$ are unknown, then they are supposed to be equal and, similar to $\overline{W}^{III}$ (see Remark 4), we have: $W^{II,k} = \frac{1}{K}, k \in J_K$ and (4.15) becomes:

$$G^a = \frac{1}{K} \sum_{k \in J_K} G^{ak}. \tag{4.17}$$

Edge Information Utilization Similar to the association network G^a construction we build the weighted network G^w combining the auxiliary normal-

ized networks $\overline{G}^w = \{G^{wk}\}_{k \in J_{K'}}$ related to the edge attributes $\{AT^{ek}\}_{k \in J_{K'}}$:

$$\omega(G^{wk}) = 1,\ k \in J_{K'}. \tag{4.18}$$

The networks family $\overline{G}^w$ is built using the same network structure as G:

$$\forall k \in J_{K'},\ V(G^{wk}) = V,\ E(G^{wk}) = E. \tag{4.19}$$

The One Discrete Edge Attribute Network G^{wk} Formation Similar to merging nodes with the same values of attributes into attribute clusters of the node partition $\mathcal{AC}^k$ (see (4.2)), for edges we introduce $\mathcal{EC}^k = \{EC_l^k\}_{l \in J_{L'_k}}$ - an edge set partition into edge clusters related to different AT^{ek}-values ($k \in J_{K'}$). Here $EC_l^k \subseteq E$ is an *edge attribute cluster* that share value at_l^{ek} of the edge attribute AT^{ek}. Similar to (4.4) for the edge attribute clusters (ECs) we have:

$$EC_l^k = \{\{v_i, v_j\} \in E :\ a'^k_{\{i,j\}} = at_l^{ek}\},\ |EC_l^k| = m'^k_l\ (l \in J_{L'_k},\ k \in J_{K'}), \tag{4.20}$$

where $\sum_{l \in J_{L'_k}} m'^k_l = m$, $k \in J_{K'}$. Identical to the node partition set $\mathcal{AC}$ (see (4.3)) we can build $\mathcal{EC} = (\mathcal{EC}^k)_{k \in J_{K'}}$ - a tuple of E-partitions into ECs of different ATEs.

In the same way as the conditions (4.5)-(4.9) were used for G^{ak}, each network G^{wk}, $k \in J_{K'}$, satisfies four conditions: a) the normalization condition (4.18); b) the edge set formation condition (4.19); c) the weights uniformity within

ECs:

$$\text{if } i,i',j,j' \in J_n : \exists\, l \in J_{L'_k} : \{v_i,v_j\},\{v_{i'},v_{j'}\} \in EC^k_l \Rightarrow w^{wk}_{ij} = w^{wk}_{i'j'}, \quad (4.21)$$

and d) the proportion of edge weights within EC^k, $EC^{k'}$ to the weights of the corresponding ATEVs:

$$\frac{w^{wk}_{ij}}{w^{ak}_{i'j'}} = \frac{W'^{III,lk}}{W'^{III,l'k}}, \quad (4.22)$$

where $\overline{w}^{wk} = (w^{wk}_{ij})_{i,j\in J_n}$ is the WAM of G^{wk}, $k \in J_{K'}$.

Similarly to the network G^{ak} the conditions (4.21),(4.22) are satisfied by choice of the weights $\{w^{wk}_{ij}\}_{i,j}$ proportionally to $W'^{III,lk}$: $w^{wk}_{ij} = \nu'(k) \cdot W'^{III,lk}$ within EC^k_l, otherwise 0. Here $\nu'(k)$ is a normalized factor of G^{wk}, which is defined from (4.18) by (4.20):

$1 = \omega(G^{wk}) = \sum_{i,j} w^{wk}_{ij} = \sum_{l\in J_{L'_k}} \sum_{\{v_i,v_j\}\in EC^k_l} w^{wk}_{ij} = \sum_{l\in J_{L'_k}} \sum_{\{v_i,v_j\}\in EC^k_l} \nu'(k) \cdot W'^{III,lk} = \nu'(k) \cdot \sum_{l\in J_{L'_k}} W'^{III,lk} \cdot 2|EC^k_l| = \nu'(k) \cdot \sum_{l\in J_{L'_k}} W'^{III,lk} \cdot 2m'^k_l,$

wherefrom

$$\nu'(k) = \frac{1}{2\sum_{l\in J_{L'_k}} W'^{III,lk} \cdot m'^k_l}, \quad (4.23)$$

$$w^{wk}_{ij} = \frac{W'^{III,lk}}{\sum_{l'\in J_{L'_k}} 2W'^{III,l'k} \cdot m'^k_{l'}} \text{ if } \{v_i,v_j\} \in E, \text{ otherwise } 0. \quad (4.24)$$

Remark 5. *Analogically to nodes (see Remark 3) for edges we assume normalization of the weights* $\{\overline{W'}^{IIIk}\}_{k\in J_{K'}}$ *(see* (4.1)*).*

Remark 6. *If* $\overline{W'}^{IIIk}$ *is unknown then for edges the result similar to* (4.14)

can be obtained : a) by Remark 2 all ATEVs are equally ($W'^{III,lk} = W'^{III,l'k}$, $\forall l, l' \in J_{L'_k}$*); b) by Remark 5* $||\overline{W}'^{III,k}|| = 1$ *(see* (4.1)*, hence,* $W'^{III,lk} = \frac{1}{|\overline{W}'^{III,k}|} = \frac{1}{L'_k}$ *and the equations* (4.23),(4.24) *can be rewritten as:*

$$\nu'(k) = \frac{L'_k}{2\sum_{l\in J_{L'_k}} m'^k_l},$$

$$w^{wk}_{ij} = \frac{1}{2\sum_{l\in J_{L'_k}} m'^k_l} \text{ if } \{v_i, v_j\} \in E, \text{ otherwise 0.}$$

The Weighted Network G^w construction Similar to G^a construction we form the weighted network G^w as the weighted network sum of networks $\overline{G}^w$ with parameters $\overline{W}'^{II}$:

$$G^w = \sum_{k\in J_{K'}} W'^{II,k} \cdot G^{wk}. \tag{4.25}$$

Lemma 2. *If* (4.18) *holds then* G^w *is normalized:*

$$\omega(G^w) = 1. \tag{4.26}$$

Proof. *By* (4.1)*,* (4.18) *and* (4.25) *we obtain:* $\omega(G^w) = \omega(\sum_{k\in J_{K'}} W'^{II,k} \cdot G^{wk}) = \sum_{k\in J_{K'}} W'^{II,k} \cdot \omega(G^{wk}) = \sum_{k\in J_{K'}} W'^{II,k} = 1$*, hence the normalization condition* (4.26) *is satisfied for* G^w*.*

Due to (4.19), the vertex and edge sets are not changed during the linear network transformation from $\overline{G}^w$ into G^w, hence $V(G^w) = V$, $E(G^w) = E$; a WAM $\overline{w}^w = (w^w_{ij})_{i,j}$ of G^w is the following linear combination of the WAMs $\{\overline{w}^{wk}\}_{k\in J_{K'}}$: $\overline{w}^w = \sum_{k\in J_{K'}} W'^{II,k} \cdot \overline{w}^{wk}$.

Weights of Attributes and Attributes' Values The weights $\overline{W}^{II}$, $\overline{W}^{'II}$, $\{\overline{W}^{III,k}, \overline{W}^{'III,k}\}_k$ (see (4.1)) play an important role in forming the networks G^a, G^w. For their evaluation an expert assessment, pairwise comparisons (see Section 3.3), etc. can be used. Here we describe how to derive $\overline{W}^{II}, \overline{W}^{'III,k}$ directly from the networks similar to the HSTN.

Extracting Weights of Node Attributes from Attributed Networks For determining of the node attribute weights $\overline{W}^{II}$ we use two different ways to compare the internal degrees with the total degrees of node partitions into attribute clusters and the internal weights with the total weights of these partitions (see (2.7), (2.8)):

- First way: $\forall k \in J_K$ for the node attribute cluster partition $\mathcal{AC}^k$ find a ratio of the internal degree $d^{int}(\mathcal{AC}^k)$ and total degree $d(\mathcal{AC}^k)$, then normalize the resulting vector:

$$W_1^{II} = \frac{1}{\sum_k \frac{d^{int}(\mathcal{AC}^k)}{d(\mathcal{AC}^k)}} \left(\frac{d^{int}(\mathcal{AC}^k)}{d(\mathcal{AC}^k)} \right)_{k \in J_K}; \tag{4.27}$$

- Second way: similarly for the partitions find ratios of the internal weights $\omega^{int}(\mathcal{AC}^k)$ and total weights $\omega(\mathcal{AC}^k)$ $(k \in J_K)$, then normalising the ratios vector:

$$W_2^{II} = \frac{1}{\sum_k \frac{\omega^{int}(\mathcal{AC}^k)}{\omega(\mathcal{AC}^k)}} \left(\frac{\omega^{int}(\mathcal{AC}^k)}{\omega(\mathcal{AC}^k)} \right)_{k \in J_K}. \tag{4.28}$$

Extracting Weights of Enumerated Edge Attributes As mentioned above numerical networks attributes, such as edge weights, contain valuable information. If for some reason the quantitative information is unknown, then the networks are considered as unweighted and relevant CDAs are applied to them. Importance of using weights in CD is demonstrated on the HSTN (see Section 5.4.2), where modularity is considerably higher when edge weighs are used. But the initial HSTN data (see Section 4.2) provides us only with ranks of texting intensity (see Table 4.2).

Suppose similarly to the HSTN a network G is decorated by one edge attribute only, which is enumerated. Hence, $K' = 1$ and index l (see (2.3)) can be eliminated. The edge attribute, its values and type can be represented as follows: $\overline{AT}^e = (AT^{e1}) = (AT^e)$, $\overline{at}^{e1} = \overline{at}^e = \{at_l^e\}_{l \in J_{L'}}$, $\overline{AT}^e_{type} = (AT^{e1}_{type}) = (AT^e_{type}) = (E)$.

Problem statement We wish to transform the edge attrributed network G into the weighted network G^w.

For that we conduct a transformation of enumerated ATEVs $\overline{at}^e$ into numerated ones $\overline{at}^{we} = \{at_l^{we}\}_{l \in J_{L'}}$ and then use the last ones as G^w-edges weights. We represent it as a search of weight function:

$$y = \varphi(x) : at_l^{we} = \varphi(at_l^e)\,(l \in J_{L'}). \tag{4.29}$$

We recommend two ways for that: a) an expert assessment and b) optimization of a criterion Cr. Describe briefly them both:

- **Expert assessment** An expert, familiar with the network and with nature of relations between its nodes, can be asked: a) to assign directly the weights $\overline{at}^{we}$; b) to indicate type of the function (4.29); c) to perform pairwise comparison $\{at_l^e \backslash at_{l'}^e\}_{l \neq l'}$ following by computing a vector of priorities $\overline{at}^{we}$ (see Section 3.3). If an expert group works on this assessment, then each expert individual estimates are aggregated into the group estimate $\overline{at}^{we}$.

  This is traditional way used in Decision Theory to extract numerical data in conditions of uncertainty, when we are not able to formalise the problem clearly. Particularly, we are not able to formulate a criterion of optimization. In this case we expect that experts are able to give qualitative estimates, so that the attribute weights' estimates will be close to actual ones.

- **Modularity maximization approach** we use in robust case when we are able to formulate our goal as an optimization problem with a criterion Cr. Different functions can be chosen as the criterion Cr (see Sect.2.1). For instance, if we aim to conduct qualitative CD then modularity, conductance, etc. can be the Cr.

Suppose $Cr = M$ then the modularity maximization problem can be represented as a boolean linear program [BDG$^+$08], [Das14]:

$$M = \frac{1}{2m} \sum_{u,v \in V} (a_{u,v} - \frac{d_u d_v}{2m})(1 - x_{uv}) - \sum_{u \in V} \frac{d_v^2}{2m} \rightarrow max \tag{4.30}$$

subject to $a)\, x_{u,v} = 0$, if u, v at the same community; otherwise $x_{u,v} = 1$;

$b) \forall u \neq v \neq z :\, x_{u,z} \leq x_{u,v} + x_{v,z}$.

For weighted networks we use the function (2.23) instead of (4.30), so modularity maximization is equivalent to optimization of the following linear function:

$$M^w = \frac{1}{2 \cdot \omega(G)} \sum_{u,v \in V} \left(a^w_{u,v} - \frac{s_u s_v}{2 \cdot \omega(G)} \right) (1 - x_{uv}) - \sum_{u \in V} \frac{s_v^2}{2 \cdot \omega(G)} \rightarrow max. \tag{4.31}$$

In this case $\forall u, v \in V\, a^w_{u,v} \in \overline{at}^{we} \cup \{0\}$, so, the WAM A^w is also connected to the initial attributes $\overline{at}^e$ by function φ. Our goal is to find $\varphi(.)$, x and $\overline{at}^{we}$ so that the function of modularity $M'^w = M^w(x, \overline{at}^{we})$ is maximized. The problem $M'^w \rightarrow max$ is nonlinear because in (4.31) the WAM A^w, weight $\omega(G)$ and node strengths $\{s_v\}_v$ depend on the new variables $\overline{at}^{we}$. Thus, the mixed-boolean nonlinear problem (4.31) should be solved, which is quite complicated task.

To avoid these difficulties we propose two approaches (we refer to them as Scheme 1, Scheme 2) that is applicable for non high-scale networks, such as the HSTN. Since M^w remains unchanged if A^w-elements are changed proportionally, one value in $\overline{at}^{we}$ out of L' can be chosen arbitrary. For example, if $at^{we}_{L'} = 1$ then the rest of $L' - 1$ values $\overline{at}'^{we} = \overline{at}^{we} \backslash \{1\}$ need to be determined.

Remark 7. ***Scheme 1*** *consists in the following: a) take increasing sequences* $\overline{at}'^{e} > 0$ *if* $AT^e_{type} = R$ *or any positive sequences* $\overline{at}'^{e}$ *if* $AT^e_{type} = U$*; b) run*

CD for each of the weights combinations; c) choose the one maximizing M'^{w}. *An advantage of Scheme 1 is that it allows to find the problem solution by standard CDAs. If* L' *is large, then the choice of* $\overline{at}'^{we}$ *can be simplified by Scheme 2.*

Scheme 2 *consists in selecting function* (4.29) *from known classes of functions with a few parameters. For instance, we take the function* $y = \varphi(x) = a^1 + a^2 \cdot g(a^3 x)$ *with* $\iota = 3$ *unknown parameters. Taking into account that* $\varphi(at^e_{L'}) = 1$ *we choose numerical combinations of* $\iota - 1 = 2$ *parameters, say* a^1, a^2, *obtain* $\overline{at}'^{e}$ *and repeat steps b) and c) of Scheme 1.*

Notice that the normalization of $\overline{at}^{we}$ allows to obtain the priorities of edge attribute values: $\overline{W}'^{III} = (W'^{III,1}) = (W'^{III})$,

$$W'^{III} = \frac{\overline{at}^{we}}{||\overline{at}^{we}||}. \tag{4.32}$$

Aggregation of the networks G^a and G^w into the Aggregated Network G^{wa} In the current section we present an approach for analysis of attributed networks with discrete node attributes and arbitrary edge attributes. Suppose the association network G^a and the weighted network G^w are formed then the aggregated network G^{wa} is formed as their weighted network sum with coefficients $\overline{W}^I$:

$$G^{wa} = W^I \cdot G^a + W'^I \cdot G^w. \tag{4.33}$$

Lemma 3. *If* (4.16) *and* (4.26) *hold then* G^{wa} *is normalized:*

$$\omega(G^{wa}) = 1. \tag{4.34}$$

Proof. *In addition to* (4.16) *and* (4.26) *using the normalization condition* (4.1) *for* $\overline{W}^I$ *we have* $\omega(G^{wa}) = W^I \cdot \omega(G^a) + W'^I \cdot \omega(G^w) = W^I + W'^I = 1$, *hence* (4.34) *holds.*

Parameters of G^{wa} are the following: a vertex set $\mathrm{V}(G^{wa})$ coincides with the set V, an edge set $\mathrm{E}(G^{wa})$ is a combination of G^a- and G^w- edge sets, a WAM $\overline{w}^{wa}$ is the corresponding linear combination of the WAMs $\overline{w}^a, \overline{w}^w$. So,

$$V(G^{wa}) = V,\ E(G^{wa}) = E(G^a) \cup E(G^w),\ \overline{w}^{wa} = W^I \cdot \overline{w}^a + W'^I \cdot \overline{w}^w.$$

For the final attributed network G^{wa} construction we conduct two steps of aggregating the data of three levels. It is represented by a three-level hierarchy (see Figure 4.3.1).

As it is seen the networks associated with individual node or edge attributes from the families $\overline{G}^a$ and $\overline{G}^w$ are located on the lowest level. Depending on $\overline{W}^{II}$ and $\overline{W'}^{II}$ these networks constitute different proportions in the next upper level, which consists of the networks G^a and G^w associated with all node and edge attributes. In turn, G^a and G^w form the top level network G^{wa} and participate in the aggregated network formation to more or less extend depending on W^I, W'^I.

A generalization of the AHP synthesis phase (see Section 3.3) is based on

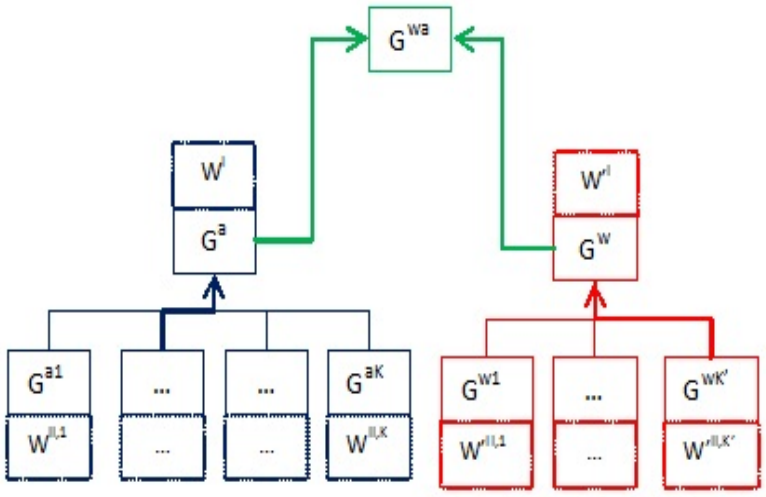


Figure 4.3: The aggregated network G^{wa} hierarchy

the observation that instead of vectors of local priorities of alternatives any other objects can be used if their linear combination is defined. For instance, they can be matrices of the same dimension, functions, networks of the same vertex set, etc. The result of the synthesis will be an object of the same type, such as a matrix of a relevant dimension, a function or a network of the same vertex set.

So, at the bottom level of the hierarchy we use the networks $\{G^{ak}, G^{wk}\}_k$ of individual node-edge attributes instead of vectors of local priorities of alternatives (see Section 3.3). Then the networks G^a, G^w and G^{wa} are built as their linear combination. $\overline{W}^{II}$, $\overline{W}'^{II}$ and $\overline{W}^{I}$ play a role of vectors of local priorities. This operation is defined since the vertex set is the same for all of the networks. In comparison with the basic AHP, where a numerical global priorities vector is the result of the synthesis phase, the outcome of

our procedure is the top level normalized network G^{wa} (see Lemma 3).

Generalization for the Continuous Node Attributes By the assumption (see Section 2.3) node attributes $\overline{AT}^n$ are discrete, and the scheme of converting such networks into the aggregated network G^{wa} is described above. But if nodes are decorated by at least one continuous attribute the method does not work. In this case for all the C-type node attributes we suggest to use a distance measure (see Section 3.1) and build also an association network G^c composed the continuous node attributes. After that all these intermediate networks G^a, G^w and G^c can be combined into a new normalized network $G^{wac} = W^I \cdot G^a + W'^I \cdot G^w + W''^I \cdot G^c$ ($W^I + W'^I + W''^I = 1$), where $W''^I > 0$ is the weight of the continuous node attribute part G^c in the accumulated network G^{wac}.

4.3.2 Attributed Networks Applications

Problem 1: Multi-Layer Community Detection in Attributed Networks In Section 4.3.1 the weighted network G^w is represented as a linear combination of auxiliary networks of individual edge attributes (see (4.25)). On the other hand, forming edge set we assumed that the edges are formed exceptionally based on node attributes, therefore CD is conducted on G^w depending on node attributes. Analogically to the associated network G^a representation (4.15) we represent G^w as a linear combination of auxiliary

networks $\overline{G} = \{G^k\}_k$ related to single node attributes:

$$G^w = \sum_{k \in J_K} W^{II,k} \cdot G^k. \tag{4.35}$$

Combine G^w and G^a into the network G^{wa} by (4.33). Now G^{wa} can be decomposed into the networks corresponding to ATNs only:

$$G^{wa} = \sum_{k \in J_K} W^{II,k} \cdot G^{wak}, \text{ where } G^{wak} = W^I \cdot G^{ak} + W'^I \cdot G^k, k \in J_K. \tag{4.36}$$

For such type of networks our approach to MLCD consists in the following: we run CD on the accumulated network G^{wa}, obtain the initial partition $\mathcal{C}^*$ (see (2.17)) and define a likely node attribute AT^{nk_0} underlying the partition (see (4.63)). Then we reduce edge weights of G^{wa} by subtracting the network G^{wak_0}, run CD again on the new network $G^{wa} - G^{wak_0}$, obtain a new node partition and define a node attribute likely underlying it, etc.

The MLCD Algorithm

- **Step 1**. Set up $\tau = 1$ - an initial iteration, $G(1) = G^{wa}$ - an initial network, $I(1) = J_K$;
- **Step 2**. Run CD in $G(\tau)$ and obtain the partition $\mathcal{C}^{*\tau}$;
- **Step 3**. Determine $k_\tau \in I(\tau)$ such that AT^{nk_τ} is the likely node attribute (LATN) of the partition $\mathcal{C}^{*\tau}$ (see (4.63));

- **Step 4**. Derive G^{wak_τ} from $G(\tau)$: $G(\tau+1) = G(\tau) - G^{wak_\tau}$, $I(\tau+1) = I(\tau)\backslash\{k_\tau\}$;

- **Step 5**. Assign $\tau = \tau + 1$. If $\tau \leq K$ then go to Step 2, otherwise stop.

The network G^k constructing In comparison with the network G^{ak} (see Observation 1), where an edge set $E(G^{ak})$ was formed independently from E, an edge set $E(G^k)$ of G^k is an intersection of E and $E(G^{ak})$. If a representation similar to (4.13) is used

$$G^k = G[\mathcal{AC}^k] = \underset{l\in J_{L_k}}{\cup} G[AC_l^k], \tag{4.37}$$

then the G^k-weight is distributed between $|E(G^k)| = \sum_{l\in J_{L_k}} m_l^k$ edges, where $m_l^k = |E(G[AC_l^k])|$, $l \in J_{L_k}$.

There are many different ways of the edge weight distribution. Ideally $\overline{G}$-networks should, first of all, satisfy (4.35). It can be rewritten in the following form:

- Condition 0:

$$w_{ij}^w = \sum_{k\in J_K} W^{II,k} \cdot w_{ij}^k,\ i,j \in J_n, \tag{4.38}$$

  where $\overline{w}^k = (w_{ij}^k)_{i,j\in J_n}$ is a WAM of G^k. Secondly, it is desirable that each G^k satisfies conditions similar to (4.5)-(4.9):

- Condition 1. the normalization:

$$\omega(G^k) = 1; \tag{4.39}$$

- Condition 2. An edge $\{v_i, v_j\} \in E(G^k)$ exists if

$$\exists l \in J_{L_k} : v_i, v_j \in AC_l^k, \{v_i, v_j\} \in E; \tag{4.40}$$

- Condition 3. The edge weights distribution - if for $i, i', j, j' \in J_n$ (4.40) holds and

$$\exists l' \in J_{L_k} : v_{i'}, v_{j'} \in AC_{l'}^k, \{v_{i'}, v_{j'}\} \in E \tag{4.41}$$

then

$$\frac{w_{ij}^k}{w_{i'j'}^k} = \frac{W^{III,lk}}{W^{III,l'k}}, \tag{4.42}$$

- Condition 4. Edge weights are equal within AC_l^k:

$$\text{if for } i, i', j, j' \in J_n \text{ (4.40), (4.41) hold and } l = l' \Rightarrow w_{ij}^k = w_{i'j'}^k. \tag{4.43}$$

It is not so easy to have a situation when all these conditions are met. The problem is that normally Conditions 0-4 do not hold simultaneously. Demonstrate it on the examples below:

1. If we require to satisfy the condition (4.38) strictly then for the each edge $\{v_i, v_j\} \in E$ we can distribute the weight w_{ij}^w between $\{w_{ij}^k\}_k$

proportionally to $W^{II,k}$ priorities of node attributes:

$$w_{ij}^{k(I)} = \frac{w_{ij}^{w}}{|B_{ij}| \cdot W^{II,k}} \text{ if } k \in B_{ij}, \text{ otherwise } 0\,(i,j \in J_n), \tag{4.44}$$

where

$$B_{ij} = \{k : a_i^k = a_j^k\} \subseteq J_K \tag{4.45}$$

is the set of indices of common ATNVs for $v_i, v_j \in V$.

2. If we require strict satisfaction of Conditions 1-4 then the weights distribution is obtained similarly to (4.11):

$$w_{ij}^{k(II)} = \nu'''(k) \cdot W^{III,lk} \text{ if (4.40) holds, otherwise } 0\,(i,j \in J_n), \tag{4.46}$$

where $\nu'''(k) = \left(\sum_{l \in J_{L_k}} 2 \cdot W^{III,lk} \cdot m_l^k\right)^{-1}$, $k \in J_K$.

If the condition (4.38) holds for each such $G^k \in \overline{G}$ then the distribution that satisfies Conditions 0-4 is found:

$$w_{ij}^{k} = w_{ij}^{k(I)} = w_{ij}^{k(II)} \; \forall i,j,k. \tag{4.47}$$

Until now, we assumed that the network is known. It is possible that we know only the network' model. In the case the network and its decomposition into $\overline{G}$ can be found based on this model (see Section 4.4).

Normally the condition (4.47) does not hold for real-world networks, because even if they follows the known model, during data collecting - nodes, edges and their attributes - errors and inaccuracies are inevitable.

Notice that during implementation of the MLCD algorithm the weight of $G(\tau)$ is reduced by W^{II,k_τ}:

$$\omega(G(\tau+1)) = \omega(G(\tau)) - W^{II,k_\tau}. \tag{4.48}$$

It means that the impact of the node attribute AT^{nk_τ} completely disappear after this iteration, the current detected layer of the network is detected, and we have a chance to see next layers of the network.

If we are not able to find the weight distribution that satisfies (4.47) then the following approaches may be used:

- **Approach 1** Relax Condition 1 and consider a network satisfying only $w_{ij}^k = w_{ij}^{k(II)}\, \forall i,j,k$ instead of (4.47).

- **Approach 2** Introduce a network G^{K+1} that accumulates an effect of unaccounted factors and attributes did not included in $\overline{AT}^n$, $\overline{AT}^e$. Then the representation (4.35) becomes

$$G^w = \mu \sum_{k \in J_K} W^{II,k} \cdot G^k + G^{K+1},$$

  where $\mu \in [0,1]$ is defined as a maximal value such that the G^{K+1}-WAM remains non-negative, $A^{w,K+1} \geq 0$:

$$\mu = \min_{ij} \frac{w_{ij}^w}{\sum_{k \in J_K} W^{II,k} \cdot w_{ij}^k}.$$

  In this case the weight of $G(\tau)$ is reduced by less than W^{II,k_τ} and

(4.48) is rewritten as $\omega(G(\tau+1)) = \omega(G(\tau)) - W^{II,k_\tau}(W^I + \mu \cdot W'^I) > \omega(G(\tau)) - W^{II,k_\tau}$.

An important part of the MLCD algorithm implementation is detecting node attributes likely underlying the set of partitions $\{\mathcal{C}^{*\tau}\}_\tau$. This part is described later in the current section.

Remark 8. *The MLCD algorithm can be implemented just for the initial network G^w without complementing it by the association network G^a. For this their weights $(W^I, W'^I) = (0, 1)$ should be chosen.*
Nevertheless, we recommend to use them both, because the goal of the networks G^w and G^a joint consideration is utilization of all the network data. If the weights are unknown then they are supposed to be equal, $W^I = W'^I = 0.5$, meaning that the attributes in $\overline{AT}^n$ are only factors of the network formation. For community detection goals we expect that adding G^a highlights partitions into communities formed under influence of node attributes.

Problem 2: Node Attribute Network Decoration Suppose we have an IIAN G^0 with partially missing node attributes. We wish to restore these attributes and obtain the corresponding CIAN G.

For the G^0-elements we use similar to G notations adding the mark 0. For example, Λ^0 is its matrix of nodes' attributes. Thus, G is represented in the form (2.14) and the IIAN - in the similar form $G^0 = (V, E, \Lambda^0, \Lambda'^0)$ where $\Lambda^0 \not> 0$, $\Lambda, \Lambda', \Lambda^{0'} > 0$ (see Section 4.1). Let $\overline{at}^{n0} = \{at^{n0,k}\}_{k \in J_K}$, $at^{n0,k} = \{at_l^{n0,k}\}_{l \in J^0_{L_k}} = \{0\} \cup at^{n,k}$ ($J^0_{L_k} = J_{L_k} \cup \{0\}$) be sets of G^0-ATNVs.

We call the attribute clusters (4.4) as *complete attribute clusters* and restore them from *incomplete attribute clusters* AC_l^{0k}:

$$AC_l^{0k} = \{v_i \in V : a_i^k = at_l^{n0,k}\},\ |AC_l^{0k}| = n_l^{0k},\ l \in J_{L_k}^0,\ k \in J_K, \tag{4.49}$$

$\sum_{l \in J_{L_k}^0} n_l^{0k} = n$, that forms a node partition $\mathcal{AC}^{0k}$ into these incomplete ACs. $\mathcal{AC}^{0k} = \{AC_l^{0k}\}_{l \in J_{L_k}^0}$. Similar to $\overline{\mathcal{AC}}$ in (4.3),

$$\overline{\mathcal{AC}}^0 = (\mathcal{AC}^{0k})_{k \in J_K} \tag{4.50}$$

forms a tuple of partitions into incomplete attribute clusters (4.49).

In the notations (4.49) we aim to restore $n^0 > 0$ node attributes, $n^0 = \sum_{k \in J_K} n_0^{0k}$, in G^0. According to the classification of Network Topology Inference Problems given in Section 3.1, this is NAIP, specifically *Network Node Attribute Inference Problem* since it supposes restoring only node attributes. Analogously to the auxiliary network G^{ak} (see Section 4.3.1), which we call a *complete association network*, we construct an *incomplete association network* $G^{0a,k}$ with edges only within the known part $\mathcal{AC}^{0k} \backslash AC_0^{0k}$ of the partition $\mathcal{AC}^k$. An unknown part AC_0^{0k} of the partition we match with a part of the incomplete association network by zero edge weights (if $v_i, v_j \in V$: $a_i^k = a_j^k = 0 \Rightarrow w_{ij}^{0a,k} = 0$, where $\overline{w}^{0a,k} = (w_{ij}^{0a,k})_{ij}$ is the $G^{0a,k}$-WAM).

The incomplete association network G^{0a} is formed similarly to (4.15):

$$G^{0a} = \sum_{k \in J_K} W^{II,k} \cdot G^{0a,k}, \tag{4.51}$$

where the auxiliary incomplete association networks $\{G^{0a,k}\}_k$ are normalized:

$$\omega(G^{0a,k}) = 1,\ k \in J_K,$$

and satisfy conditions similar to (4.6)-(4.9). For instance the edge $\{v_i, v_i\}$ existence condition (4.10) becomes

$$\exists l \in J_{L_k} :\ v_i, v_j \in AC_l^{0k}. \tag{4.52}$$

Similarly to (4.11) and (4.12) formulas derived for the complete association network G^{ak}, we have:

$$w_{ij}^{0a,k} = \nu^0(k) \cdot W^{III,lk} \text{ if (4.52) holds, otherwise } 0\,(i, j \in J_n),$$

where $\nu^0(k) = \left(\sum_{l \in J_{L_k}} W^{III,lk} \cdot n_l^{0k} \cdot (n_l^{0k} - 1)\right)^{-1}$ is a normalized factor of $G^{0a,k}$.

Due to the lack of information regarding the complete ACs, we complement G^w by the incomplete association network (4.51) instead of the complete association network G^a (see (4.15)). The aggregation condition (4.33) becomes: $G^{0wa} = W^I \cdot G^{0a} + W'^I \cdot G^w$.

An idea of our approach to Network Node Attribute Inference Problem solving is that instead of the attributed network G^{wa} we apply CD to the network G^{0wa}, which is incomplete since it combines the complete weighted network G^{w} with the incomplete association network G^{0a}. When a result of CD, the $\mathcal{C}^*$ partition, is obtained then: a) we determine k^* - an index of a likely node attribute of $\mathcal{C}^*$; b) $\forall C \in \mathcal{C}^*$ we derive node attributes value $at_{l_C}^{k^*} \in \{at_l^{nk^*}\}$ likely underlying the community C; c) we split the cluster $AC_0^{0k^*}$ into subclusters, $AC_0^{0k^*} = \cup_{C \in \mathcal{C}^*} AC_{0C}^{0k^*}$, and assign to nodes $v \in AC_0^{0k^*}$ missing values of AT^{nk^*} from the $\{at_{l_C}^{k^*}\}$ depending on communities, to which the node v belongs; d) finally, we reduce the edge weights in G^{0wa} subtracting from it a weighted network sum of G^{0a,k^*} and G^{wk^*}, run CD again and so on.

An algorithm that we call the Multi-Layer Node Attribute Inference Algorithm (MLNI algorithm) was briefly described before. It was represented as a sequence of operations applied to networks in the same manner as it was done in the MLCD algorithm.

Similar to (4.36) the network G^{0wa} is decomposed into K subnetworks corresponding the individual node attributes:

$$G^{0wa} = \sum_{k \in J_K} W^{II,k} \cdot G^{0wa,k}, \text{ where } G^{0wa,k} = W^{I} \cdot G^{0ak} + W^{'I} \cdot G^{k}, k \in J_K. \tag{4.53}$$

Notice that the networks $\{G^{0wa,k}\}_k$ are unknown at the beginning because the decomposition (4.35) is unknown (see (4.37)). This decomposition is formed iteratively during the MLNI algorithm implementation:

The MLNI Algorithm

- **Step 1**. Set up $\tau = 1$ - an initial iteration, $G(1) = G^{0wa}$ - an initial network, $I(1) = J_K$;
- **Step 2**. Run CD for $G(\tau)$ and obtain the $\mathcal{C}^{*\tau}$ partition;
- **Step 3**. Determine $k_\tau \in I(\tau)$ such that AT^{nk_τ} is a LATN of the partition $\mathcal{C}^{*\tau}$ (see (4.63));
- **Step 4**. $\forall C \in \mathcal{C}^{*\tau}$ find a C-LATNV $at_{l_C}^{n,k_\tau}$ (see (4.61),(4.62)) and assign it to $\forall v_i \in AC_0^{0k_\tau}$, $a_i^{k_\tau} = at_{l_C}^{n,k_\tau}$;
- **Step 5**. Based on the restored $\{AC_l^k\}_l$ form the partition $\mathcal{AC}^k$ and the network G^k according to (4.37);
- **Step 6**. Form G^{0wa,k_τ} in (4.53) and derive it from $G(\tau)$: $G(\tau+1) = G(\tau) - G^{0wa,k_\tau}$, $I(\tau+1) = I(\tau)\backslash\{k_\tau\}$;
- **Step 7**. Assign $\tau = \tau + 1$. If $\tau \leq K$ go to Step 2, otherwise stop.

The LPA modifications for the Network Node Attribute Inference Problem The LPA with seeds (see Sections 3.2.1-3.2.2) can be applied for Network Node Attribute Inference Problem if incomplete ACs are used as seeds.

The LPA modification 1:

- Step 0. Fix $k \in J_K$, choose L_k seeds $AC_l^{0k}, l \in J_{L_k}$ (see (4.49));

- Step 1. Assign unique labels to each of the seeds, say $1 - L_k$, and nodes $x \in AC_0^{0k}$;
- Step 2. Establish an order of iterative revising $x \in AC_0^{0k}$;
- Step 3. Each node $x \in AC_0^{0k}$ is revised in this order, and a new label is assigned to it according to (3.1). If the choice is not unique then the LPA Breaking Ties Rule is applied;
- Step 4. The process continues until labels change;
- Step 5. The restored node attribute clusters $AC_l'^k \supseteq AC_l^{0k}, l \in J_{L_k}$, correspond to communities with labels $1 - L_k$.

Notice that if $\sum_{l \in L_k} |AC_l'^k| = n$ then $AC_l^k = AC_l'^k, l \in L_k$, the partition $\mathcal{A}C^k$ is found, and Network Node Attribute Inference Problem is solved completely for the node attribute AT^{nk}, otherwise partially.

The advantage of the modification is that it uses as nodes' attribute information, particularly about AT^{nk}, as the network structure and propagation properties of the network. As disadvantages the following can be listed: a) there is no guarantee that the attribute clusters in $\mathcal{A}C^k$ are completely restored; b) the available information regarding the rest of node attributes and their weights is not used, hence if a dominant partition exists then even an existence of a "good" initial state, given by $\mathcal{A}C^{0k}$ and corresponding to the AT^{nk}, can be not helpful for restoring the partition $\mathcal{A}C^k$ by the LPA modification.

We can overcome the first drawback if we apply a LPA Modification 2 choosing new labels among the labels of the seeds. Respectively, (3.1) becomes $l_x^{new} = \underset{l \in J_{L_k}}{argmax} \sum_{u \in V} a_{ux} \delta(l_u, l)$. The Breaking Ties Rule is modified for the case where $x \in AC_0^{0k}$ is not adjacent to seeds - a x-label is simply chosen randomly from $1 - L_k$. The second drawback can be overcome partially by using G^{wa} instead of G^w.

The MLNI algorithm and the LPA modification 2 can be used together. For example in Step 4 of the MLNI algorithm, where for all missing attributes there are assigned the same values within $C \in \mathcal{C}^*$, we can use node labels obtained by the LPA Modification 2. So, the MLNI Step 4 modification looks like: $\forall C \in \mathcal{C}^*$ run the LPA Modification 2 within C expanding only labels $1 - L_k$ presented in the C.

Problem 3: Decoration of Communities We analyse the partition $\mathcal{C}^*$ (see (2.17)) from different sides defining: a) node attributes values causing forming a community $C \in \mathcal{C}^*$ (Problem 3.1); b) a main node attribute causing formation of the community C (Problem 3.2); c) a main node attribute causing the partition $\mathcal{C}^*$ formation in the whole (Problem 3.3).

For Problem 3.1 we fix $k \in J_K$ and combine the attribute clusters $AC_l^k \in \mathcal{AC}^k$ (see (4.3)) with the communities in $C \in \mathcal{C}^*$ (see (2.17)). We obtain $L_k \cdot L^*$ nodes clusters:

$$AC_{ll*}^k = AC_l^k \cap C_{l^*},\ n_{ll^*}^k = |AC_{ll*}^k|,\ l \in J_{L_k},\ l^* \in J_{L^*}. \tag{4.54}$$

We evaluate the following proportion of the number of nodes in the ACs (see (4.4)):

$$p_l^{0k} = \frac{n_l^k}{n}, l \in J_{L_k}. \tag{4.55}$$

The proportion (4.55) is compared with the proportion of the number of nodes in the clusters (4.54) to the cardinality of (4.4):

$$p_{ll*}^k = \frac{n_{ll*}^k}{n_{l^*}}, l \in J_{L_k},\ l^* \in J_{L^*}. \tag{4.56}$$

We choose a pre-specified level of significance $\alpha \in (0,1)$. By the One Proportion Test $\forall l \in J_{L_k}$, $l^* \in J_{L^*}$ we test the hypothesis: $H_{ll*}^{0,k} = \{p_{ll*}^k = p_l^{0k}\}$ versus $H_{ll*}^{1,k} = \{p_{ll*}^k > p_l^{0k}\}$. The test statistic is

$$z_{ll^*}^k = \frac{p_{ll^*}^k - p_l^{0k}}{\sqrt{p_l^{0k}(1-p_l^{0k})}}\sqrt{n_{ll^*}^k},\ l \in J_{L_k},\ l^* \in J_{L^*}, \tag{4.57}$$

which, by assumption, follows the standard normal distribution.

If for l', l'^* the null hypothesis $H_{l'l'*}^{0,k}$ is rejected then in the community $C_{l'*}$ the proportion of the $AC_{l'}^k$-nodes is significantly higher. So, with the level of significance α we justified that the community $C_{l'*}$ can be decorated by the value $at_{l'}^{nk}$ of AT^{nk}:

$$at^{C_{l'*},k} = at_{l'}^{nk}. \tag{4.58}$$

If (4.58) holds then we can say that nodes with $at_{l'}^{nk}$-value of AT^{nk} underlie the community $C_{l'*}$. In other words the value $at_{l'}^{nk}$ is the *underlying ATNV* (UATNV) of the community $C_{l'*}$. When this $at_{l'}^{nk}$-search is applied for $k \in J_K$,

a number of attributes values underlying the $C \in \mathcal{C}^*$ can be found.

Normally the result of such communities decoration by underlying ATNVs is assigning the attributes to only part of communities and node attributes. It implies that the rest of the communities attributes are missing (the same as the attributes take zero-values). Let

$$I_{l*} = \{k \in J_K : at^{C_{l^*},k} > 0\},\ l^* \in J_{L^*}, \tag{4.59}$$

denote sets of indices of assigned attribute values to $C_{l*} \in \mathcal{C}^*$.

We can increase $|I_{l*}|$ by increasing α, thereby decreasing the probability $p = 1 - \alpha$ of the results justification. Here we face the following problem - if high α is chosen then to a part of the communities can be assigned more than one attribute value of the same node attribute.

On the other hand, occasionally we need to assign unique node attributes to each community (see, for instance, the MLNI algorithm, Step 4). The choice can be done in different ways, for instance, using the deviation $p^k_{ll^*} - p^{0k}_l$. We recommend to use observed values of the test statistic (4.57) and take maximum by l:

$$l_{k,l^*} :\ z^k_{l_{k,l^*}l^*} = \underset{l \in J_{L_k}}{max}\, z^k_{ll^*},\ l^* \in J_{L^*}. \tag{4.60}$$

If (4.60) is applied to each community $C_{l^*} \in \mathcal{C}^*$ and to each node attribute $AT^{nk} \in \overline{AT}^n$ then the number $L^* \cdot K$ of node attribute values is obtained. We call them node attribute values likely underlying corresponding communities

or simply *likely ATNVs* (LATNVs) of the communities and denote them $lat^{C,k}$ similar to (4.58):

$$lat^{C_{l^*},k} = at^{nk}_{l_{k,l^*}}. \tag{4.61}$$

This is a way of solving Problem 3.1 without justification of the result. Nevertheless it can be useful (see the MLNI algorithm).

In Step 3 of the MLCD and MLNI algorithms we use an index of the node attribute likely underlying a node partition. Let k^* denote the index for a partition into communities $\mathcal{C}^*$. Analogically to LATNVs the attribute AT^{nk^*} we call *likely underlying* the partition $\mathcal{C}^*$ *ATN* (the partition $\mathcal{C}^*$-LATN).

We will solve Problem 3.2 and then use its solution for Problem 3.3.

Problem 3.2: $\forall C_{l^*} \in \mathcal{C}^*$ find maximum of (4.61) and the corresponding node attribute:

$$k'_{l^*} : z^{k'_{l^*}}_{l_{k,l^*}l^*} = \max_{k \in J_K} z^{k}_{l_{k,l^*}l^*}. \tag{4.62}$$

The attribute $AT^{nk'_{l^*}}$ will be the ATN likely underlying the community $C_{l^*} \in \mathcal{C}^*$ (the community C_{l^*}-LATN). If the choice (4.62) is not unique we make random choice from the candidates.

Then a tuple

$$\overline{AT}^* = (AT^{nk'_{l^*}})_{l^* \in J_{L^*}}$$

that satisfies (4.62) is a tuple of the $\mathcal{C}^*$-communities likely ATNs (the partition $\mathcal{C}^*$ LATNs).

Let L^* be a set of different indices of LATNs: $I^* = S(\{k'_{l^*}\}_{l^* \in L^*})$. Problem 3.3:

among $AT^{nk} \in \overline{AT}^*$ find those AT^{nk^*} that accumulate most of the nodes:

$$k^* : n^* = \sum_{l^* \in J_{L^*}} n_{l^*} \delta_{k'_{l^*} k^*} = \max_{k \in I^*} \sum_{l^* \in J_{L^*}} n_{l^*} \delta_{k'_{l^*} k}. \tag{4.63}$$

We can also obtain in such way the community $C_{l^*} \in \mathcal{C}^*$ *underlying ATN* (the community C_{l^*}-UATN) and the partition $\mathcal{C}^*$ underlying ATN ($\mathcal{C}^*$-UATN) if in Problem 3.2 solution will use the ATNs from (4.59). Then for the community C_{l^*}-UATN determination ($C_{l^*} \in \mathcal{C}^*$) a formula is obtained from (4.62):

$$k''_{l^*} : z^{k''_{l^*}}_{l_{k,l^*} l^*} = \underset{k \in I_{l^*}}{max}\, z^{k}_{l_{k,l^*} l^*}. \tag{4.64}$$

Then the partition $\mathcal{C}^*$-UATN is derived from (4.63):

$$k^{**} : n^{**} = \sum_{l^* \in J_{L^*}} n_{l^*} \delta_{k''_{l^*} k^{**}} = \max_{k \in I^{**}} \sum_{l^* \in J_{L^*}} n_{l^*} \delta_{k''_{l^*} k}, \tag{4.65}$$

where $I^{**} = S(\{k''_{l^*}\}_{l^* \in L^{**}})$ is a set of different UATNs indices.
And finally,

- if an unique LATN is assigned to all communities in $\mathcal{C}^*$, $|S(I^*)| = 1$, we say that the AT^{nk^*} is the *likely dominant node attribute* of the partition C^*;

- if an unique UATN is assigned to $\mathcal{C}^*$-communities:

$$|S(I^{**})| = 1, \tag{4.66}$$

we say that the $AT^{nk^{**}}$ is the *dominant node attribute* of the partition $\mathcal{C}^*$.

We say that the *result* of the MLCD or MLNI is *justified* with probability p if it is obtained with use of underlying ATNs and ATNVs with level of significance $\alpha = 1 - p$.

Remark 9. *If instead of a CIAN we consider an IIAN then the same scheme is applicable with only difference that* (4.54) *becomes*

$$AC^k_{ll*} = AC^{0k}_l \cap C_{l^*},\ n^k_{ll^*} = |AC^k_{ll*}|,\ l \in J^0_{L_k},\ l^* \in J_{L^*}$$

and (4.55) *becomes*

$$p^{0k}_l = \frac{n^{0k}_l}{n}, l \in J^0_{L_k}.$$

Remark 10. *if* $\forall\, k \in J_K$*:* $n^k_l = n^k$, $l \in J_{L_k}$ *then for determining LATNVs and LATNs, instead of comparing the test statistics* (4.57) *in* (4.61),(4.62)*, the frequencies* $n^k_{ll^*}$ *(see* (4.54)*) can be compared.*

4.4 Human Communication Network Models

At this section we touch formation of attributed networks (see Problem 4 in Chapter 1). We are wondering how an attributed network (2.14) is formed if the information about vertices V and their attributes Λ is known. In other words we review formation of an edge set E and its attributes Λ' that we refer as a Problem 4.1 and a Problem 4.2, respectively.

The only assumption we make, also used in Section 4.3, is that the links are formed due to similarity of node's attributes.

4.4.1 Problem 4: Attributed Networks Formation

We present several ways to solve Problem 4.2 and form an edge set of an attributed network if node attributes are known. For that several attributed network models will be presented. For convenience we interpret the models in terms of communication between people related to common activities/interests (AIs). In the networks nodes are people and their AIs are attributes of the nodes.

Model 1 - an association network model An association network G^a (see (4.15)) is an example of an attributed network where links exist between nodes that have at least one the same attribute. It can be interpreted as a network of virtual contacts between people with the same interests. If two persons have a common interest then they are connected in some way. Supporting of such contact does not need anything.

To each activity/interest (AI) we correspond a network G^{ak}, and two networks G^{ak}, $G^{ak'}$ $(k \neq k')$ are formed independently. In terms of (4.13) $G^w = G^a$ can be represented as follows: it is a weighted network sum (4.35) of K networks, which are collections of complete graphs:

$$G^k = G^a[\mathcal{AC}^k] = \bigcup_{l \in J_{L_k}} K_{n_l^k}, k \in J_K. \tag{4.67}$$

Thus, the network is overlapping of K partitions by disjoint union of complete graphs. Weights of the network are defined by (4.11),(4.35).

Model 2 - an attributed networks model based on Erdös-Rényi Model. Suppose for the existing link a similarity of node attributes is necessary but not enough because of randomness. Similar to Model 1 we represent the network G^w by (4.35). Links in the additional networks G^{wk} are created randomly with probability p_l^k between two nodes v_i, v_j sharing the value at_l^{nk} of the attribute AT^{nk}. Hence, G^k is a node partition by Erdös-Rényi Random Graphs (ERRGs) [ER59]:

$$G^k = G^w[\mathcal{AC}^k] = \underset{l \in J_{L_k}}{\cup} ERRG(p_l^k, n_l^k), k \in J_K. \tag{4.68}$$

The resulting network G^w is overlapping of K partitions by ERRGs. Edge weights of each subnetworks G^{ak} are defined by (4.46).

In terms of human communication Model 2 simulates a real situation when a group of people is formed simultaneously. Contacts of each user occur randomly without analysing any prior information due to its inaccessibility. The communication can be established on a regular basis only if these people actually have common interest. Different type contacts are formed independently.

Model 3 - an attributed networks model based on Barabási-Albert Model. In comparison to Model 2, in the current model we review a situation when a group of people is formed gradually. First of all, group members

aspire to contacts with popular and authoritative colleagues in each area of the expertise. First these contacts are formed for the most important AIs and then for the less important ones, and a chance to clarify common interests is higher if the contact already exists.

As before the network G^w is a weighted network sum (4.35) of subnetworks G^k related to single node attributes ($k \in J_K$). The networks $\{G^k\}_k$ are formed consecutively by k according to their priorities. In terms of (4.1) it means that $W^{II,k} \geq W^{II,k+1}$, $k \in J_{K-1}$. For each k the edge set of G^k is formed between vertices with the same value of AT^{nk} consecutively by $i \in J_n$ with probabilities depending on degrees $\{d^k_{i'}\}_{i'}$ of all preceding nodes $v_{i'}$ ($i' < i$) and parameters p^k, p'^k ($p^k \leq p'^k$) for new and previously established contacts, respectively.

There are many ways of a generalisation for attributed networks of Barabási-Albert Preferential Attachment Model [AB02]. For instance each network G^k related a single node attribute is formed as follows: disjoint subsets of nodes are connected by preferential attachment and then the obtained graphs are connected and form the whole node set partition $\mathcal{AC}^k$. These all partition are united into a multi-layer cover $\mathcal{AC}$ with respect to node attributes priorities, values in Λ and pre-assigned order of the vertices arising. The network layers are dependent regardless we consider the case $p^k = p'^k$, $k \in J_K$ (Model 3.1) or the case where $\exists k \in J_K : p^k < p'^k$.

Model 3.1 simulates a node partition by Barabási-Albert Graphs (BAGs).

Each G^k can be represented in a manner of (4.67),(4.68) as following:

$$G^k = G^w[\mathcal{AC}^k] = \bigcup_{l \in J_{L_k}} BAG(n_l^k, \alpha_l^k), k \in J_K, \tag{4.69}$$

where α_l^k is the power of preferential attachment in AC_l^k, $l \in J_{L_k}$, $k \in J_K$.

4.4.2 Human Communication Model Assumptions

This section is dedicated to Problem 4.2. Two models presented in Section 4.4.1 - Model 2 and Model 3 - can simulate networks of real, natural contacts, implying requirements to spend time or another resource for their supporting. When these attributed networks are simulated and edge weights are assigned then we can solve Problems 1-3. In Section 4.3.1 it was described one way to distribute weights according to priorities of networks elements. Here we present one more way related to natural people communication. We refer to the network with the corresponding edge weight distribution as a Human Communication Network (HCN). If an edge set of the network is formed according to Model 2 (see Section 4.4.1) then we call the network model as the *HCN Model 2*, if the set is created according to Model 3 (see Section 4.4.1) then the model is called the *HCN Model 3*.

Suppose the network's edge set E is formed according to Model 2 or Model 3 presented in Section 4.4.1. To specify edge weights in the corresponding HCN Model 2 or the HCN Model 3, first some natural assumptions and restrictions on human communication are introduced. Then the model is formalized and

explicit formulas for the weights of the network $\overline{G}^w$ are derived. It makes the model convenient for applying the MLCD and MLNI algorithms (see Sec.4.3.2).

- **Condition 1**. People's AIs are formed outside of the network (exogenous data);
- **Condition 2**. Connections between people are possible only if they have common AIs;
- **Condition 3**. Each person distributes uniformly time for the AI AT^{nk} between friends with the same interest (the time t^k for supporting one contact related to the AI AT^{nk} is known, $k \in J_K$);
- **Condition 4**. For everyone possibility of the communication is restricted by time T. If for a person the time is not enough for supporting his/her contacts then the time allotted for supporting one contact related to the AT^{nk} and $AT^{nk'}$ ($AT^{nk'} \neq AT^{nk'}$) is distributed proportionally to t^k and $t^{k'}$;
- **Condition 5**. If two persons with the same interest are ready to devote time to each other then, if it is necessary, they come to a compromise following certain rules.

Unite t^k, $k \in J_K$ into a tuple $\overline{t}$:

$$\overline{t} = (t^k)_{k \in J_K}. \tag{4.70}$$

Formalise Conditions 1-5 in terms of a WAM $A^w = (a^w_{ij})_{ij}$ of the weighted network G^w. We rewrite (4.35) in the form:

$$G^w = \sum_{k \in J_K} G^{W^{II}k}, \text{ where } G^{W^{II}k} = W^{II,k} \cdot G^k, k \in J_K. \tag{4.71}$$

In comparison with (4.35), the formula (4.71) represents G^w as a sum of K subnetworks related to individual ATNs, which are not normalized.

Introduce networks G^*, G'^* and G^w satisfying Conditions 1-3, 1-4 and 1-5, respectively. Let $A^* = (a^*_{ij})_{ij}$ and $A'^* = (a^{'*}_{ij})_{ij}$ be the WAMs of G^* and G'^*, respectively.

Similarly to the (4.71), we represent the networks G^* and G'^* as networks sums: $G^* = \sum_{k \in J_K} G^{*k}$, $G'^* = \sum_{k \in J_K} G'^{*k}$ (G^{*k}, G'^{*k} are subnetworks of G^* and G'^* related to AT^{nk}). Respectively, the WAMs $A^w, A^* = (a^*_{ij})_{ij}, A'^* = (a^{'*}_{ij})_{ij}$ of these networks are represented as follows:

$$A^w = \sum_{k \in J_K} A^{W^{II}k}, A^* = \sum_{k \in J_K} A^{*k}, A'^* = \sum_{k \in J_K} A'^{*k}, \tag{4.72}$$

where $A^{W^{II}k} = (a^{W^{II}k}_{ij})_{ij}, A^{*k} = (a^{*k}_{ij})_{ij}, A'^{*k} = (a^{'*k}_{ij})_{ij}$ are WAMs of $G^{W^{II},k}$, G^{*k}, G'^{*k}, respectively ($k \in J_K$).

First we form the network G^*. We build the networks G'^* and G^w from it adding Condition 4 and Condition 5 consecutively.

Let the set B_{ij} of v_i, v_j common ATNVs be found by (4.45). Then $N^k_i = \{v_j \in N_i : v_j \in B_{ij}\}$ is a set of v_i-neighbours with the same ATNVs as

v_i. Expand the notations of the node degree and node strength from the N_i-set into sets $\{N_i^k\}_k$: $d_i^k = |N_i^k|$ is the node attribute AT^{nk}-degree of v_i; $s_i^{wk} = \sum_{v_j \in N_i^k} a_{ij}^{WIIk}$, $s_i^{*k} = \sum_{v_j \in N_i^k} a_{ij}^{*k}$, $s_i^{\prime *k} = \sum_{v_j \in N_i^k} a_{ij}^{\prime *k}$ are AT^{nk}-strengths of v_i in G^w, G^*, G'^*, respectively ($i \in J_n$, $k \in J_K$). Then the strength of v_i in G^w, G^* or G'^* is $s_i^w = \sum_k s_i^{wk}$, $s_i^* = \sum_k s_i^{*k}$ or $s_i'^* = \sum_k s_i'^{*k}$, correspondingly.

1. Start with the G^*-edge weights distribution:

   (a) **Condition 1** implies that the network G^* is decorated by discrete, $\overline{AT}^n$, or continuous, $\overline{AT}^c$, node attributes and their values are given, hence the matrix Λ in (2.20) is known;

   (b) **Condition 2** means that the links are formed only by similarity of the node's attributes. For instance, for attributed networks with discrete node attributes only ($K > 0$, $K'' = 0$, see Section 4.1) the condition looks like this: if $\forall k \in J_K \; a_i^k \neq a_j^k \Rightarrow \{v_i, v_j\} \notin E$.

   (c) **Condition 3** says that for the network G^* the condition similar to Condition 3 for G^{ak} (see Section 4.3.1) holds: if $\exists i, i', j, j' \in J_n$, $k, k' \in J_K$, $l, l' \in J_{L_k} : v_i, v_j \in AC_l^k$, $v_{i'}, v_{j'} \in AC_{l'}^{k'}$ then

$$\frac{a_{ij}^{*k}}{a_{i'j'}^{*k'}} = \frac{t^k}{t^{k'}}. \tag{4.73}$$

   Since there is no restriction on the communication time for G^*, it implies that all of the contacts are supported at the appropriate level. So, assigned weights are equal to the maximal needed time

t^k. Then

$$a^*_{ij} = \sum_{k \in J_K} a^{*k}_{ij} = \sum_{k \in B_{ij}} t^k,\ i, j \in J_n. \tag{4.74}$$

(4.74) says how edge weights in G^*, G^{*k} are defined. Notice that the matrix A^* is symmetric, hence the network G^* is undirected.

Communication time of the each person depends on the number of the contacts of each type, and therefore the strengths of G^*-nodes are defined as the following:

$$s^*_i = \sum_{j \in J_n} a^*_{ij} = \sum_{k \in J_K} \sum_{j \in J_n} a^{*k}_{ij} = \sum_{k \in J_K} s^{*k}_i = \sum_{k \in J_K} d^k_i t^k,\ i \in J_n. \tag{4.75}$$

In terms of the model values s^*_i, s^{*k}_i can be interpreted as a time that a person i could potentially devote for the communication overall and for the particular AI correspondingly.

2. Moving on to the network G'^* we add Condition 4 - the time restriction - to the network G^*. This condition determines how much time a person i is ready to spend for supporting each AI AT^{nk}-contact depending on his/her priorities and the number of contacts.

   Condition 4 can be expressed as a restriction on node strengths by T-value: $s'^*_i \leq T, \forall i \in J_n$. If $s^*_i \leq T, \forall i \in J_n$ then it is true that $G'^* = G^*$, otherwise the weights a^{*k}_{ij} are scaled to meet the time restriction:

$$a'^{*k}_{ij} = \nu^*_i a^{*k}_{ij},\ i, j \in J_n, \tag{4.76}$$

where the scaling parameters $\{\nu_i^*\}_i$ depend on the strengths s_i^{*k}:

$$\nu_i^* = min\left(1, \frac{T}{s_i^*}\right), \forall i. \tag{4.77}$$

Substituted (4.76) in (4.73) we obtain $\frac{a_{ij}^{'*k}}{a_{i'j'}^{'*k'}} = \frac{t^k \cdot \nu_i^*}{t^{k'} \cdot \nu_{i'}^*}$, from which it is clear that the condition similar to (4.73) is not satisfied for G'^*. The proportion is true only for each of the nodes: if for $i \in J_n \, \exists j, j' \in J_n$, $k, k' \in J_K$, $l, l' \in J_{L_k} : v_i, v_j \in AC_l^k$, $v_i, v_{j'} \in AC_{l'}^{k'}$ then

$$\frac{a_{ij}^{*k}}{a_{ij'}^{*k'}} = \frac{t^k}{t^{k'}}. \tag{4.78}$$

It means that each person distributes his/her own time independently from each other and following common priorities

$$W^{II,k} = \frac{t^k}{||t||}, \, k \in J_K. \tag{4.79}$$

Find the WAM of G'^* by (4.76):

$$a_{ij}^{'*} = \sum_{k \in J_K} a_{ij}^{'*k} = \nu_i^* \sum_{k \in J_K} a_{ij}^{*k} = \nu_i^* \cdot a_{ij}^*, \, i, j \in J_n. \tag{4.80}$$

The weight $a_{ij}^{'*}$ determines how much time a person i is ready to devote for a person j. Respectively, a weight $a_{ji}^{'*}$ describes a vice versa situation. It is clear that normally these values do not coincide, $a_{ij}^{'*} \neq a_{ji}^{'*}$. So, the network G'^* is directed, which does not display natural human

communication. To describe the real situation we consider constructing the final network G^w from G'^*.

3. By adding Condition 5 the abstract directed network G'^* is transformed into the undirected network G^w, which weights $\{a^w_{ij}\}_{ij}$ are equal to time that both persons - i and j - actually devote to each other. They are posteriori weights, which are obtained as a result of a compromise between two persons, who are ready to spent together not the same time.

   Consider two persons i and j having contacts ($B_{ij} \neq \{\emptyset\}$) who are looking for a compromise since $a'^*_{ij} \neq a'^*_{ji}$. The result of their common decision can be expressed as function of these weights $a^w_{ij} = f(a'^*_{ij}, a'^*_{ji})$. Without loss of generality, assume that $a'^*_{ij} \leq a^{*'}_{ji}$ and as function $f(.)$ choose linear one:

$$a^w_{ij} = \alpha \cdot a'^*_{ij} + \beta \cdot a'^*_{ji},\ i, j \in J_n, \tag{4.81}$$

   where $\alpha, \beta \geq 0$, $\alpha + \beta = 1$. (4.81) is the formula for assigning edge weights in G^w. The obtained WAM A^w is symmetric, hence the network G^w is undirected.

The function (4.81) depending on the α, β-parameters expresses different real situations:

- if $(\alpha, \beta) = (1, 0)$ then the function is $a^w_{ij} = min(a'^*_{ij}, a'^*_{ji})$ meaning that

these two persons spend together only the time they both can afford for that. The function represents majority of real situations when the communication is not obligatorily. It means that $s_i^{'*} \leq T, i \in J_n$, so, some people can not use all the time resource;

- the situation where $(\alpha, \beta) = (0, 1)$ corresponds to the function $a_{ij}^w = max(a_{ij}^{'*}, a_{ji}^{'*})$ meaning that to satisfy the communication request is in a high priority. This function describes the situation when one of the two persons has quite limited circle of the AI-friends and is very sensitive to the others attention. It can be related to kids and parents, students and teachers, and so on. In this case a person for the communication with another one has to devote more time than he/she expected at the beginning using some extra time ($s_i^{'*} \geq T, i \in J_n$);

- α, β may be defined for each pair i, j individually. For instance, if $(\alpha, \beta) = (\frac{s_i^*}{s_i^*+s_j^*}, \frac{s_j^*}{s_i^*+s_j^*})$ then the $f(.)$ takes into account a point of view of the more authoritative person. Such function can describe relationships between a professor and a student: if a professor is ready to devote time for a student, disregard how busy the student is, he should accept it.

- our approach is illustrated for the case of "fair" compromise, when $a_{ij}^{'*}, a_{ji}^{'*}$ are averaged:

$$a_{ij}^w = \frac{1}{2}(a_{ij}^{'*} + a_{ji}^{'*}),\ i, j \in J_n. \tag{4.82}$$

In this case for the communication people use in average time T.

Substitute (4.80) with (4.82) and use symmetry of A^*: $a^w_{ij} = \frac{1}{2}(\nu^*_i a^*_{ij} + \nu^*_j a^*_{ji}) = \frac{1}{2}(\nu^*_i a^*_{ij} + \nu^*_j a^*_{ij})$, $i, j \in J_n$, wherefrom

$$a^w_{ij} = a^*_{ij}\frac{\nu^*_i + \nu^*_j}{2},\ i, j \in J_n.$$

According to (4.26) the network G^w is normalized and by (2.4) we obtain an adjacency matrix $\overline{w}^w$ of its normalization $\frac{G^w}{\omega(G^w)}$:

$$w^w_{ij} = \frac{a^*_{ij}}{2||A^w||}(\nu^*_i + \nu^*_j),\ i, j \in J_n. \tag{4.83}$$

4.4.3 HCNs applications

Now we review how to apply the HCN model for the two problems considered in Section 4.3.2. These are Problem 1 - community detection - and Problem 2 - network node attributes restoring. From (4.83),(4.74), and (4.72) we have: $w^w_{ij} = \frac{1}{2||A^w||}(\nu^*_i + \nu^*_j)\sum_{k\in J_K} a^{*k}_{ij} = \frac{1}{2||A^w||}(\nu^*_i + \nu^*_j)\sum_{k\in B_{ij}} t^k$ or by (4.79)

$$w^w_{ij} = \frac{||t||}{2||A^w||}(\nu^*_i + \nu^*_j)\sum_{k\in B_{ij}} W^{II,k}. \tag{4.84}$$

From (4.84) it is clear how G^w is decomposed into $\{G^{W^{II},k}\}_k$: $w^w_{ij} = \sum_{k\in J_K} a^{W^{II},k}_{ij}$, where

$$a^{W^{II},k}_{ij} = \frac{||t||}{2||A^w||}(\nu^*_i + \nu^*_j) \text{ if } k \in B_{ij}, \text{ otherwise } 0. \tag{4.85}$$

Let $A^k = (a^k_{ij})_{ij}$, $A^{wk} = (a^{wk}_{ij})_{ij}$ be an adjacency matrix and a WAM of the network G^k, respectively ($k \in J_K$). Use (4.71) and (4.85) to represent G^w as a weighted network sum (4.35) of $\{G^k\}_k$. For that $\overline{w}^w$ is represented as the corresponding linear combination of $\{A^{wk}\}_k$. First rewrite (4.84) with the help of the adjacency matrix elements: $w^w_{ij} = \sum_{k\in J_K} a^{W^{II},k}_{ij} = \frac{||t||}{2||A^w||}(\nu^*_i + \nu^*_j)\sum_{k\in J_K} W^{II,k} a^k_{ij}$. Represent the last expression in the form:

$$w^w_{ij} = \sum_{k\in J_K} W^{II,k} a^{wk}_{ij},\ i,j \in J_n, \tag{4.86}$$

where

$$a^{wk}_{ij} = \frac{||t||}{2||A^w||}(\nu^*_i + \nu^*_j)a^k_{ij},\ k \in J_K. \tag{4.87}$$

(4.86) can be also rewritten as following:

$$a^{wk}_{ij} = \tfrac{||t||}{2||A^w||}(\nu^*_i + \nu^*_j) \text{ if } k \in B_{ij}, \text{ otherwise } 0.$$

In Section 4.3.2 ideal networks $\{G^k\}_k$ were described and it was shown that generally not all the "ideal" conditions (4.38)-(4.43) hold. In the case of the HCN model the first issue is a normalization of G^k - the condition (4.39) can be not satisfied. That is why, in comparison with (4.38), in (4.86) we use the elements of the WAMs $\{A^{wk}\}_k$ instead of the WAMs of normalized networks $\{\frac{G^k}{\omega(G^k)}\}_k$.

Easy to verify that for the HCN model the only condition (4.86) similar to (4.38) is satisfied for sure. For instance, the condition (4.43) says that any people with the same attitude to a particular AI spend the same time

together for the AI, which is wrong for the HCNs. The condition (4.42) says that, regardless the network participants, a time together for any two AIs is proportional to priorities of these particular AIs (see (4.78)), which can be wrong too because of possibility of a compromise.

Notice that the weight a_{ij}^{wk} of the edge corresponding to a particular AI AT^{nk} depends on many parameters of the network G^k and the whole G^w: a) on the available time T and time t^k for the AI; b) on the number of the contacts of each type for the both sides of the contacts.

That is why if $K > 1$, implying that the HCN is node multi-attributed, the HCN is an example of a multi-layer attributed network with dependent layers. The fact that we decomposed the network into single layer subnetworks allows to apply the MLCD and MLNI to the HCNs.

Chapter 5

Experimental Part

This part of the thesis is conducted with the help of popular softwares for Network Analysis - Gephi and IGraph. If a CDA is not specified then it means that we use the Louvain method (see Section 2.4), which is implemented in both listed programs. We perform series of 5 identical experiments. If the result is not stable we make a note like "The following result was obtained in 80% of the cases".

5.1 Attributed Network Simulation

The attributed network Models 1-3 presented in Section 4.4.1 we demonstrate by examples of simulations in the IGraph. The first one is a simulation of an association network G^a (Model 1), the second one demonstrates Model 2 (a network G^{wII}) and the last one - Model 3 (a network G^{wIII}).

Parameters that are common for these networks include: the order $n = 60$,

the number of the node attributes $K = 3$, the nodes are divided randomly into $\{L_k\}_k = \{5, 4, 6\}$ attribute clusters of the same sizes $(n_l^k)_{l\in J_{L_k}, k\in J_K} = ((n^k)^{L_k})_{k\in J_K} = (12^5, 15^4, 10^6)$.

Examples of Models 1-3 networks Attributed values $\{at_l^{nk}\}_{l\in J_{L_k}}$ are equally and weights of the ATNs are $\overline{W}^{II} = (0.5, 0.3, 0.2)$.

Example 1 - Model 1 simulation G^a is the weighted network sum of node partitions by $5, 4, 6$ complete graphs, respectively (see (4.35), (4.67)): $G^a = 0.5 \cdot G^1 + 0.3 \cdot G^2 + 0.2 \cdot G^3$, where, for example, $G^1 = \cup_{l\in J_5} K_{12}$. In Figure 5.1 we can see G^1, G^2 and G^3 networks, which are partitioned by disjoint union of complete graphs, and the resulting association network G^a.

Example 2 - Model 2 simulation According to (4.35) and (4.68) G^{wII} is the following weighted network sum of node partitions by random graphs: $G^{wII} = 0.5\cdot G^1 + 0.3\cdot G^2 + 0.2\cdot G^3$. The result of the simulation with parameters $(p_l^k)_{l\in J_{L_k}, k\in J_K} = ((p^k)^{L_k})_{k\in J_K} = (0.3^5, 0.3^4, 0.5^6)$ is shown in Figure 5.2. Here, for example, $G^1 = \cup_{l\in J_5} ERRG(p^1, n^1) = \cup_{l\in J_5} ERRG(0.3, 12)$ and G^{wII} is overlapping of three such partitions by random graphs.

Example 3 - Model 3 simulation Similarly for the network G^{wIII}, which is simulated according to Model 3.2, we chose a linear preferential attachment model $((\alpha_l^k)_{l\in J_{L_k}, k\in J_K} = (\alpha)_{l\in J_{L_k}, k\in J_K}, \alpha = 1)$. G^{wIII} is formed by (4.35), (4.69) and shown in Figure 5.2. Here, for instance, $G^1 = \cup_{l\in J_5} BAG(12, 1)$ is

a disjoint union of Barabási-Albert Graphs, while the resulting G^{wIII} is a connected graph that is overlapping of interconnected graphs formed from G^1, G^2, G^3 depending of node attributes of consecutively appearing vertices.

5.1.1 Attributed networks Models Comparison

The goal of the analysis is to determine if the presented attributed networks are similar to social networks. For this purpose we study main properties of social networks through the simulated networks (see Section 2.2).

For each of the simulated networks we perform CD. Modularity values along with a number of main numerical network characteristics are presented in Tables 5.1-5.3, respectively. Looking at the modularity we can see that it is pretty high for the auxiliary networks $G^1 - G^3$, $M \in [0.747, 0.833]$, while for the resulting networks it decreases essentially due to their overlaps, $M \in (0.393, 0.640)$. Only for Model 3 that is based on preferential attachment modularity is still high ($M = 0.640$) and indicates an existence of clear community structure. An illustration of these results presents in Figures 5.7-5.9, where there are shown outcomes of CD in G^a, G^{wII}, and G^{wIII}. We can see that for the first two networks a community structure is not observable, whereas for the last one, G^{wIII}, it is obvious that communities exist.

Densities of G^a, G^{wII}, and G^{wIII} are slightly less than the sum of $G^1 - G^3$ densities. The association network is the most dense ($\delta(G^a) = 0.481$), the network G^{wIII} is the most sparse ($\delta(G^{wIII}) = 0.082$) and G^{wII} is sparse too ($\delta(G^{wII}) = 0.252$). The ASPL for these networks is small, near 1.6, an

i	stat.	G^1	G^2	G^3	G^a
1	n	60	60	60	60
2	m	330	420	270	851
3	δ	0.186	0.237	0.153	0.481
4	$\bar{l}$	1	1	1	1.519
5	$\overline{CC}$	1	1	1	0.567
6	M	0.8	0.75	0.833	0.393

Table 5.1: Model 1 - G^a numerical characteristics

i	stat.	G^1	G^2	G^3	G^{wII}
1	n	60	60	60	60
2	m	128	214	154	446
3	δ	0.072	0.121	0.087	0.252
4	$\bar{l}$	1.7	1.507	1.441	1.783
5	$\overline{CC}$	0.346	0.522	0.534	0.34
6	M	0.798	0.747	0.832	0.403

Table 5.2: Model 2 - G^{wII} numerical characteristics

exception is the network based on preferential attachment, $\bar{l}(G^{wIII}) = 3.975$. At the same time, the ACC in G^{wIII} is high enough ($\overline{CC}(G^{wIII}) = 0.476$) when in G^{wII} it is lower ($\overline{CC}(G^{wII}) = 0.34$).

The last part of the analysis is a comparison of a degree distribution and a power-low distribution. In Figures 5.4-5.6 we can see that the degree distributions of the final networks G^a and G^{wII} are very different from the degree distribution of G^1 as well as from the power-low one. Only the Model 3 network has a distribution similar to the power-law one (see Figure 5.6). From two bottom plots in Figure 5.6 it is clear that the right tail of the combined network G^{wIII} is "havier" in comparison with G^1. The two bottom plots show an approximation of a logarithm of the degree distribution by a linear function. There are values of the coefficient of determination $R^2(G^1) = 0.646$ and $R^2(G^{wIII}) = 0.761$ that confirm good approximation by exponential

i	stat.	G^1	G^2	G^3	G^{wIII}
1	n	60	60	60	60
2	m	55	56	54	145
3	δ	0.031	0.032	0.031	0.082
4	$\bar{l}$	2.091	2.776	2.407	3.975
5	$\overline{CC}$				0.476
6	M	0.8	0.786	0.833	0.64

Table 5.3: Model 3 - G^{wIII} numerical characteristics

stat.	G^a	G^{wII}	G^{wIII}
δ	0	0.5	1
$\bar{l}$	1	1	0
$\overline{CC}$	1	0	0.5
M	0.5	1	0.5
power-law	0	0	1
Total	2.5	2.5	3

Table 5.4: Comparison of Models 1-3

function in the networks based on preferential attachment, especially in the combined network G^{wIII}. The distribution can be also the power-law degree distribution.

Conclusion. The attributed network G^{wIII} is the closest to a social network since it has the highest total rank of fitting main features of social networks (see Table 5.4), where 0 is the worst rank and 1 is the best rank. An accumulation of several partitions by Barabási-Albert graphs makes the aggregated network closer to a social network than the original partitions by Barabási-Albert graphs.

In terms of people communication the networks G^{wII} and G^{wIII} simulate extreme situations where all their contacts are established randomly or they all are created intentionally, respectively. But real-life communication networks most likely look like something intermediate. Thus, we propose a new model as

a mixture of the listed attributed network models $G^w = W^2 G^{wII} + W^3 G^{wIII}$. The network G^w may take advantages of both G^{wII} and G^{wIII} networks. For example, the diameter becomes low due to a presence of the subnetwork G^{wII} and the average clustering coefficient becomes higher due to the presence of G^{wIII}.

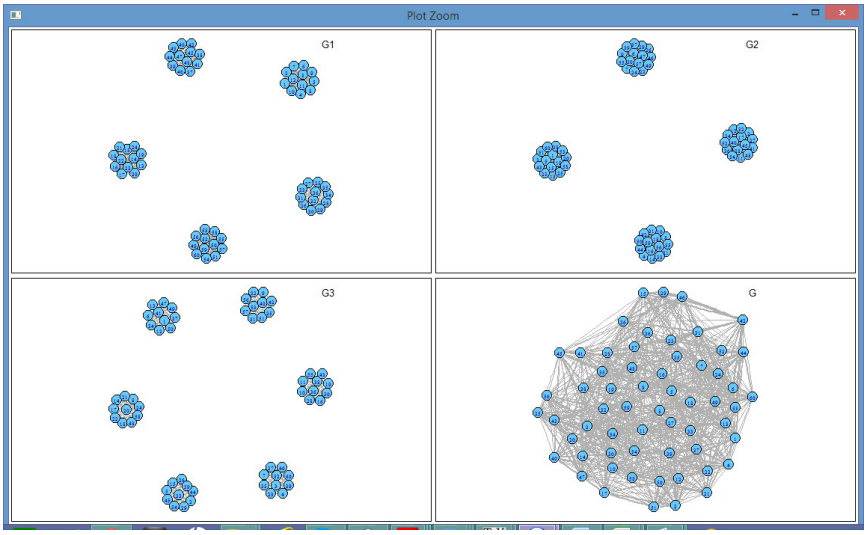


Figure 5.1: Model 1 - the association network G^a and its layers $G^1 - G^3$

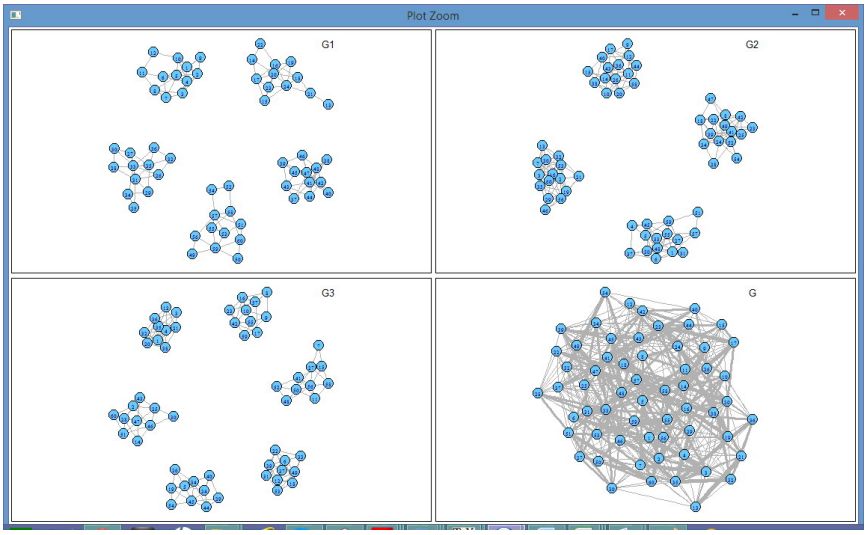


Figure 5.2: Model 2 - the weighted network G^{wII} and its layers $G^1 - G^3$

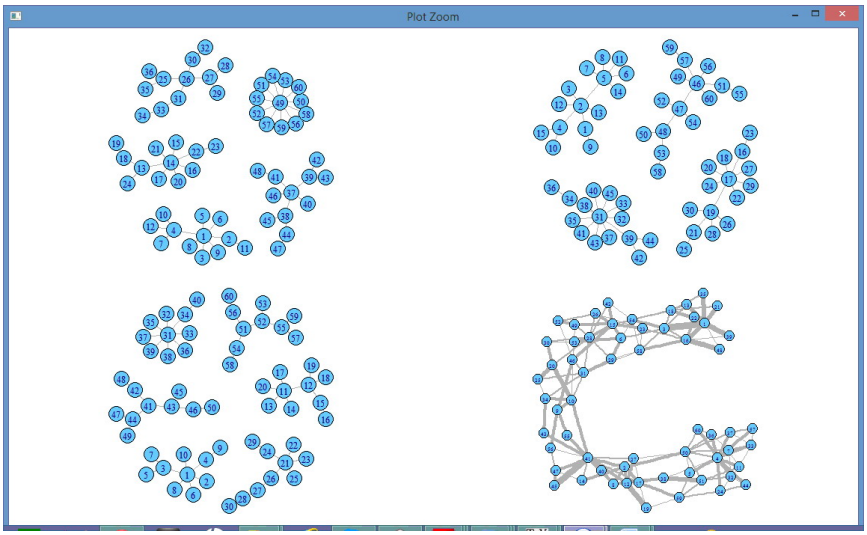


Figure 5.3: Model 3 - the weighted network G^{wIII} and its layers $G^1 - G^3$

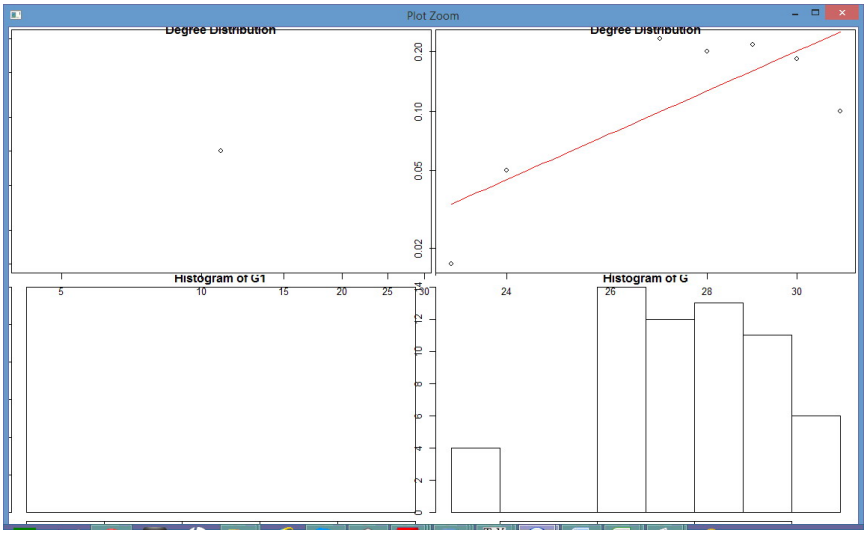


Figure 5.4: Bottom - degree distribution of G^1 (left), G^a (right). Top - comparison with exponential distribution for G^1 (left), G^a (right) in log scale

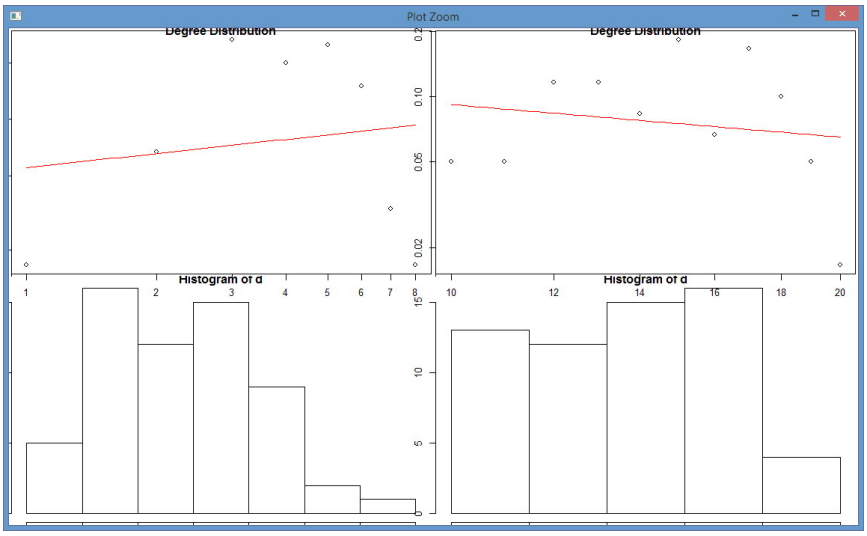


Figure 5.5: Bottom - degree distribution of G^1 (left), G^{wII} (right). Top - comparison with exponential distribution for G^1 (left), G^{wII} (right),log scale

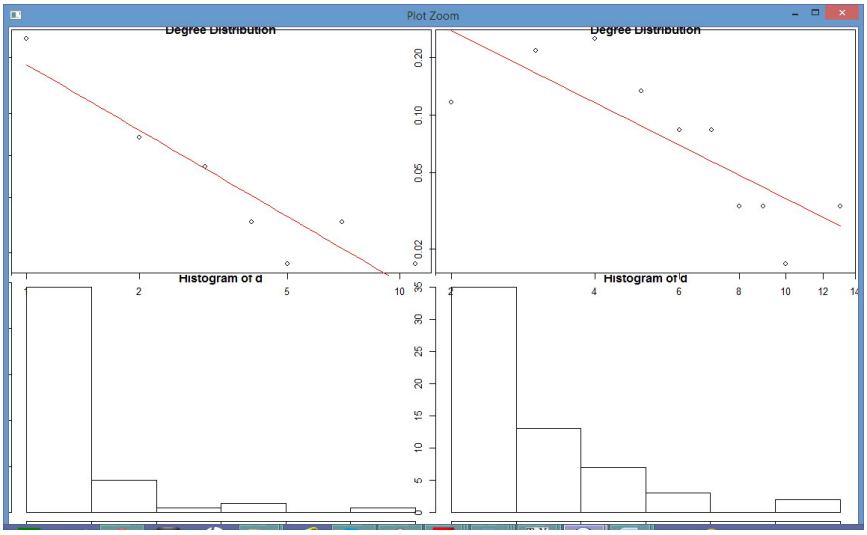


Figure 5.6: Bottom - degree distribution of G^1 (left), G^{wIII} (right). Top - comparison with exponential distribution for G^1 (left), G^{wIII} (right),log scale

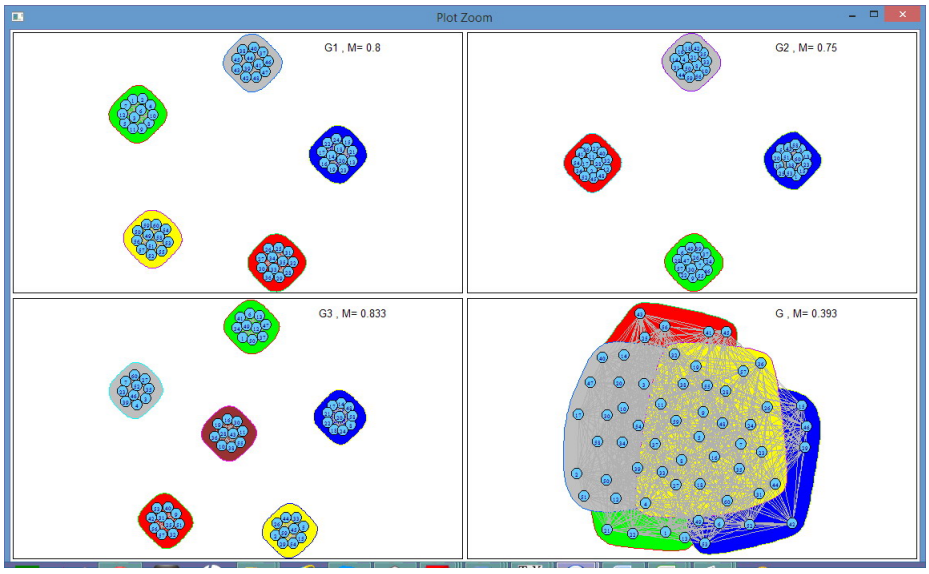


Figure 5.7: Model 1 - modularity in G^a and its layers $G^1 - G^3$

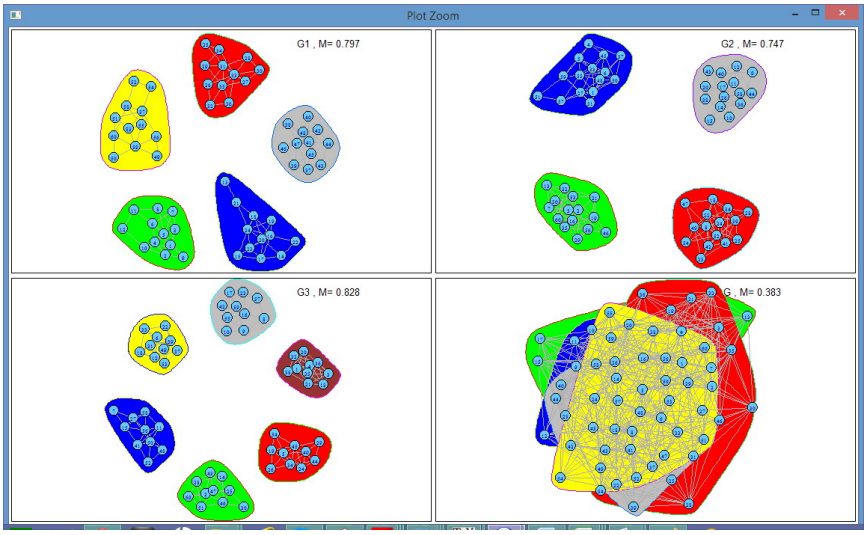


Figure 5.8: Model 2 - modularity in G^{wII} and its layers $G^1 - G^3$

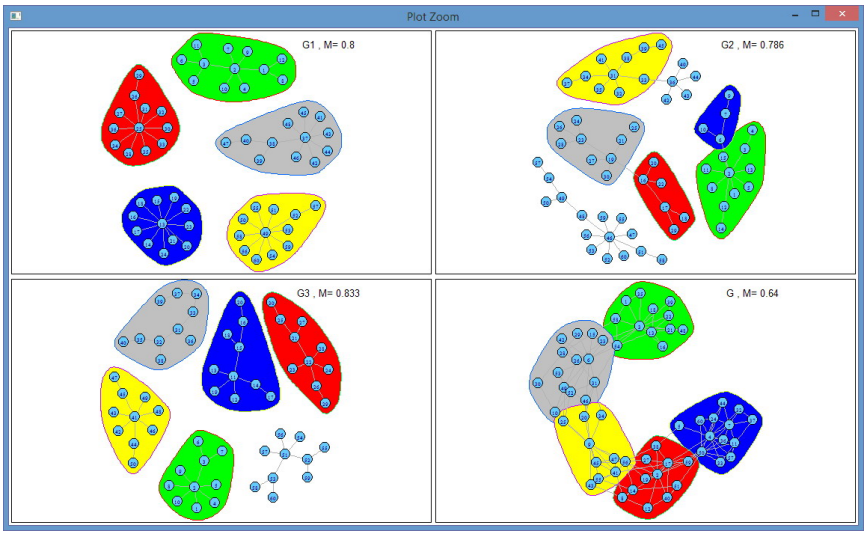


Figure 5.9: Model 3 - modularity in G^{wIII} and its layers $G^1 - G^3$

5.2 Attributed Network Applications

The effectiveness of the MLCD and MLNI algorithms (see Section 4.3.2) is demonstrated with the help of the network G^{wII} generated in Section 5.1 for the level of significance $\alpha = 90\%$.

1. We start with the MLCD algorithm. Suppose that G^{wII} is an original network for which we have information about three attributes. The result of CD in G^{wII} is 5 communities, $M = 0.383$ (see Figure 5.8), which in fact do not relate directly to any of the attributes - communities are mixtures of nodes with different attributes. On the other hand, we know that these attributes are just factors of the network formation and also we know weights of the each attribute - $0.5, 0.3, 0.2$, respectively. The only question here is an identification of these 3 layers in the network. According to the MLCD algorithm, we accompany G^{wII} by the corresponding association network G^a (see Figure 5.1) and construct an aggregated network (see Section 4.3.1). By Remark 8 $(W^I, W'^I) = (0.5, 0.5)$, $G^{wa} = 0.5G^a + 0.5G^{wII}$, $\omega(G^a) = \omega(G^{wII}) = 1$. We run the MLCD algorithm in $G(1) = G^{wa}$. The result is presented in Table 5.5. As it is seen we obtain the exact partition into attribute clusters related to the first node attribute: $\mathcal{C}^{*1} = \mathcal{C}^* = \mathcal{AC}^1$. Since in each partition $\mathcal{AC}^k$ node attribute clusters are of the same size we use Remark 10 and compare $\{n^k_{ll^*}\}_{l,l^*,k}$ in Tables 5.5-5.8. $AT^{nk^*} = AT^{n1}$ is the partition $\mathcal{C}^{*1}$-likely underlying node attribute ($\mathcal{C}^{*1}$-LATN), moreover it is also a $\mathcal{C}^{*1}$-UATN,

$k^{**} = k^* = 1$, and the dominant node attribute of the partition according to (4.66), $I^* = I^{**}$, $|S(I^{**})| = |S(\{1\})| = 1$ (see Table 5.5). An illustration provided in Figures 5.10 - 5.11 shows that attribute clusters related to both AT^{n1} and AT^{n2} can be detected. On the other hand, the weight of the AT^{n1} is $0.5/0.3 = 1.67$ times more than the weight of the AT^{n2}. Running CD, as expected, we obtain $\mathcal{C}^{*1}$ related to the AT^{n1}. The result of the next, $\tau = 2$, iteration in the network $G(2)$ of the weight $\omega(G(2)) = 1 - W^{II,1} = 0.5$ is presented in Table 5.6 and Figure 5.12. Similar to the previous step we obtain a node partition related to a new dominant node attribute AT^{n2}, $I^* = I^{**} = \{2\}$: $\mathcal{C}^{*2} = \mathcal{AC}^2$. In Figures 5.12-5.13 we can clearly see division of the network into 4 communities that are definitely related to the attribute AT^{n2}. The last iteration is conducted in $G(3)$, $\omega(G(3)) = 0.2$, and provides obtaining the partition $\mathcal{C}^{*3} = \mathcal{AC}^3$.

2. To demonstrate the MLNI algorithm we delete randomly $5, 3, 10$ attributes of nodes (that is in average 10%) and construct an incomplete association network (4.51), G^{0a}. After that we combine it with G^{wII} and construct the incomplete accumulated network $G^{0wa} = 0.5G^{0a} + 0.5G^{wII}$ (see (4.53)). Then we conduct the first iteration of the MLNI in $G(1) = G^{0wa}$. The result is presented in Table 5.7 and is very similar to the first iteration of the MLCD algorithm in G^{wa} - we obtain the node partition related to the dominant AT^{n1}: $\mathcal{C}^{*1} = \mathcal{AC}^{01}$. A visualisation is presented in Figures 5.14-5.15 - the partition $\mathcal{AC}^{01}$ into communities

can be detected visually whiles the attribute clusters in $\mathcal{AC}^{02}$ can not. There are 5 missing values of the attribute AT^{n1} and their renovation results are in Table 5.7: the first attribute value $a_{i1} = 4$ is assigned to three of the nodes, $a_{i1} = 3$ - to one more node, and $a_{i1} = 5$ - to the last one. All the restored attributes are compared to the original ones and the quality of the renovation is 100%. Except restoring missing data this step also allows to extract G^1 from G^w (see (4.35)) and $G^{0wa,1}$ from G^{0wa} (see (4.53)). The result of $\tau = 2$ iteration of the MLNI algorithm is presented in Table 5.8. In Figures 5.16-5.17 we can see four well defined communities, which surprisingly correspond to a combination of the both node attributes AT^{n2} and AT^{n3}, but neither AT^{n2} nor AT^{n3} separately. So, in the reviewed case, when we delete inconsiderable number of the node attributes, the node partitions into incomplete attribute clusters $\mathcal{AC}^{02}, \mathcal{AC}^{03}$ lead to a formation of communities, which are composed from these two. In Table 5.8 it is shown that the choice of LATNVs (4.61) is not unique. Nevertheless, by (4.63) a LATN for the $\mathcal{C}^{*2}$-partition is $AT^{k^*} = AT^2$ and 3 missing values of this attribute were restored (see Table 5.8) from the partition. Since this choice is not justified with $p = 90\%$, quality of the step renovation is only 50%.

l^*	$n^1_{1l^*}$	$n^1_{2l^*}$	$n^1_{3l^*}$	$n^1_{4l^*}$	$n^1_{5l^*}$	$n^2_{1l^*}$	$n^2_{2l^*}$	$n^2_{3l^*}$	$n^2_{4l^*}$
0	12	0	0	0	0	4	5	2	1
1	0	0	12	0	0	1	5	1	5
2	0	0	0	0	12	2	3	6	1
3	0	12	0	0	0	5	2	2	3
4	0	0	0	12	0	3	0	4	5

l^*	$n^3_{1l^*}$	$n^3_{2l^*}$	$n^3_{3l^*}$	$n^3_{4l^*}$	$n^3_{5l^*}$	$n^3_{6l^*}$	l_{1,l^*}	l_{2,l^*}	l_{3,l^*}	$n^1_{l_{1,l^*}l^*}$	$n^2_{l_{2,l^*}l^*}$	$n^3_{l_{3,l^*}l^*}$
0	1	4	1	2	3	1	1	2	2	12	5	4
1	3	1	4	3	1	0	3	2,4	3	12	5	4
2	4	2	2	1	1	2	5	3	1	12	6	4
3	0	2	2	0	3	5	2	1	6	12	5	5
4	2	1	1	4	2	2	4	4	4	12	5	4
									Total	60	26	21

Table 5.5: The MLCD $\tau = 1$-step results

l^*	$n^2_{1l^*}$	$n^2_{2l^*}$	$n^2_{3l^*}$	$n^2_{4l^*}$	$n^3_{1l^*}$	$n^3_{2l^*}$	$n^3_{3l^*}$	$n^3_{4l^*}$	$n^3_{5l^*}$	$n^3_{6l^*}$	l_{2,l^*}	l_{3,l^*}	$n^2_{l_{2,l^*}l^*}$	$n^3_{l_{3,l^*}l^*}$
0	15	0	0	0	1	4	0	2	0	8	1	6	15	8
1	0	15	0	0	5	3	1	3	3	0	2	1	15	5
2	0	0	15	0	2	2	5	2	2	2	3	3	15	5
3	0	0	0	15	2	1	4	3	5	0	4	5	15	5
												Total	60	23

Table 5.6: The MLCD $\tau = 2$-step results

l^*	$n^1_{1l^*}$	$n^1_{2l^*}$	$n^1_{3l^*}$	$n^1_{4l^*}$	$n^1_{5l^*}$	$n^2_{1l^*}$	$n^2_{2l^*}$	$n^2_{3l^*}$	$n^2_{4l^*}$	$n^3_{1l^*}$	$n^3_{2l^*}$	$n^3_{3l^*}$
0	12	0	0	0	0	4	4	0	4	1	1	1
1	0	12	0	0	0	2	3	3	3	2	0	3
2	0	0	0	0	11	2	4	5	1	2	5	0
3	0	0	11	0	0	2	2	5	2	1	1	2
4	0	0	0	9	0	4	2	2	3	2	3	3
Total	12	12	11	9	11	14	15	15	13	8	10	9

l^*	$n^3_{4l^*}$	$n^3_{5l^*}$	$n^3_{6l^*}$	l_{1,l^*}	l_{2,l^*}	l_{3,l^*}	$n^1_{l_{1,l^*}l^*}$	$n^2_{l_{2,l^*}l^*}$	$n^3_{l_{3,l^*}l^*}$	$n_{k'_{l^*}l^*}$	renew	n_{l*}
0	3	0	2	1	1,2,4	4	12	4	3	12	0	12
1	1	2	2	2	2,3,4	3	12	3	3	12	0	12
2	2	3	0	5	3	2	11	5	5	11	1	12
3	1	2	3	3	3	6	11	5	3	11	1	12
4	2	1	1	4	1	2	9	4	3	9	3	12
Total	9	8	8			Total	55	21	17	55	5	60

Table 5.7: The MLNI $\tau = 1$-step results

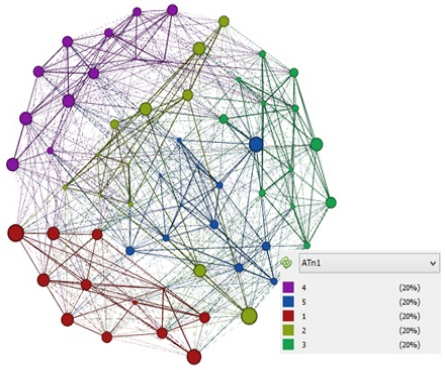


Figure 5.10: The MLCD result in $G(1)$ - $\mathcal{AC}^1 = \mathcal{C}^{*1}$

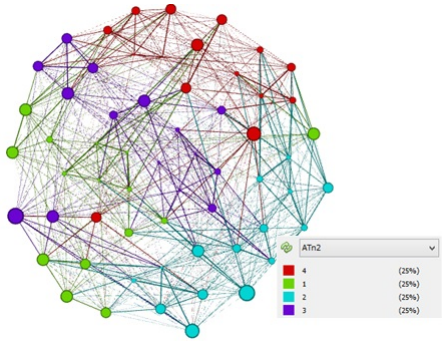


Figure 5.11: The MLCD result in $G(1)$ - $\mathcal{AC}^2$

l^*	$n^2_{1l^*}$	$n^2_{2l^*}$	$n^2_{3l^*}$	$n^2_{4l^*}$	$n^3_{1l^*}$	$n^3_{2l^*}$	$n^3_{3l^*}$	$n^3_{4l^*}$	$n^3_{5l^*}$	$n^3_{6l^*}$
0	4	4	3	3	3	1	2	3	1	1
1	3	5	4	3	1	3	1	2	3	2
2	4	4	2	3	2	3	3	3	3	0
3	3	2	6	4	2	3	3	1	1	5
Total	14	15	15	13	8	10	9	9	8	8

l^*	l_{2,l^*}	l_{3,l^*}	$n^2_{l_{2,l^*}l^*}$	$n^3_{l_{3,l^*}l^*}$	$n_{k'_{l^*}l^*}$	renew	n_{l*}
0	1,2	1,4	4	3	14	1	15
1	2	2,5	5	3	15	0	15
2	1,2	2,3,4,5	4	3	13	2	15
3	3	6	6	5	15	0	15
Total		Total	19	14	57	3	60

Table 5.8: The MLNI $\tau = 2$-step results

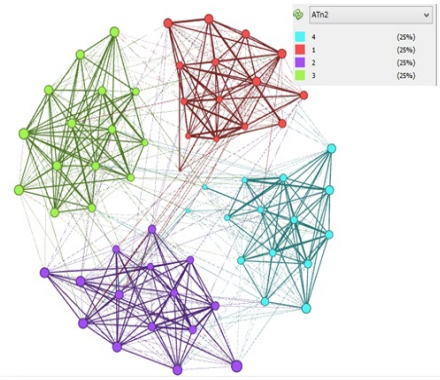


Figure 5.12: The MLCD result in $G(2)$ - $\mathcal{AC}^2 = \mathcal{C}^{*2}$

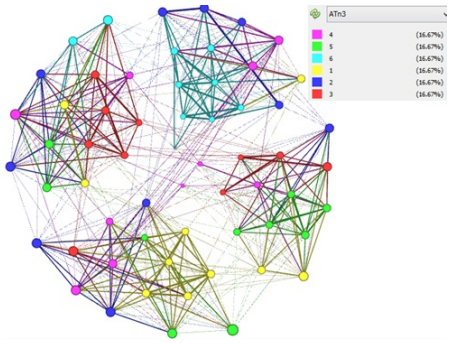


Figure 5.13: The MLCD result in $G(2)$, $\mathcal{AC}^3$

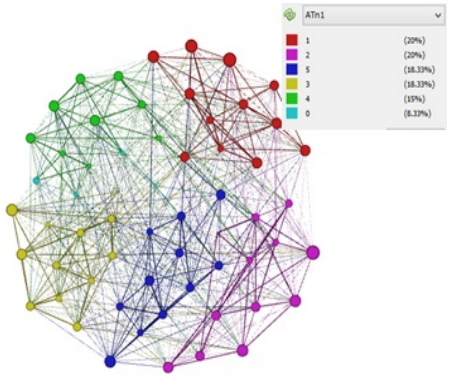


Figure 5.14: The MLNI result in $G(1)$ - $\mathcal{AC}^{0,1} = \mathcal{C}^{*1}$

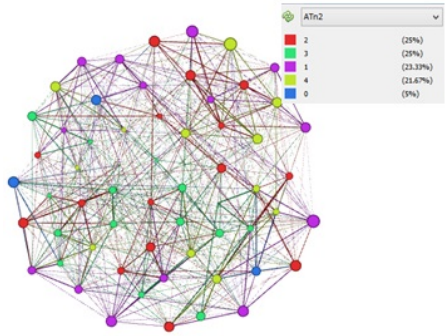


Figure 5.15: The MLNI result in $G(1)$ - $\mathcal{AC}^{0,2}$

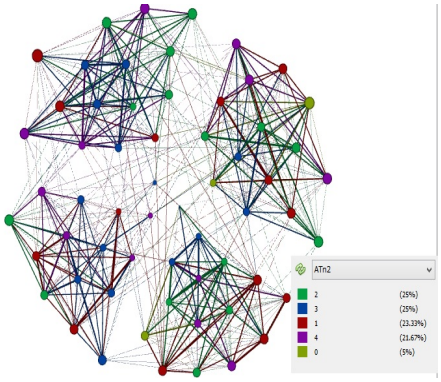


Figure 5.16: The MLNI result in $G(2)$ - $\mathcal{AC}^{0,2}$

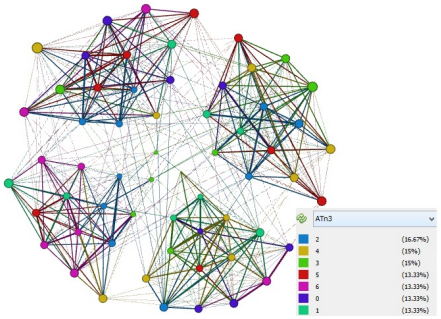


Figure 5.17: The MLNI result in $G(2)$ - $\mathcal{AC}^{0,3}$

Conclusion. From the listed examples we can conclude that: a) both algorithms are effective; b) complementing an original network by an association network improves quality of CD if we aim to detect partitions related to node attributes; c) the MLNI algorithm is sensitive to a percentage of missing attributes and belonging the corresponding nodes to node attribute clusters; d) an assignment of missing node attributes based on underlying node attribute values of particular communities is effective if choice of these values is justified (see (4.58), (4.65)). Otherwise different approaches such as the LPA-modifications (see Section 4.3.2) for Network Node Attribute Inference Problem may be useful.

5.3 Human Communication Network Simulation and Application

Similar to Section 5.2 for the HCN model demonstration we chose the network G^{wII} based on partitions by random graphs. We took the same network G^{wII} and converted it into the HCN Model 2 network (see Section. 4.4.2) by assigning weights according to (4.86). Two values of the time resource $\overline{T} = \{T_j\}_{j=1,2}$ and the vector (4.70) - $\bar{t} = (t_1, t_2, t_3) = (4, 3, 2)$ - are used. $\{t_k\}_k$ are sorted in descending order by, for example, priority of supporting the AI AT^{n1}. Extracting the vector of priorities of all AIs (4.79) we have $W^{II} = \frac{(4,3,2)}{||(4,3,2)||} = (0.45, 0.33, 0.22)$. As a result we construct the networks $G^{wII,1}, G^{wII,2}$ related to T_1, T_2, respectively. These two values are chosen in the following way: a) in the network $G^{wII,1}$ for majority, 80%, of people the time T_1 is sufficient to support their contacts completely; b) for the network $G^{wII,2}$ the situation is opposite - most, 80%, of people should distribute their time resource T_2. For the simulated G^{wII} these parameters are $\overline{T} = (56, 40)$.

In Figures 5.18-5.19 we can see the described HCNs and also we can observe that edge weights in $G^{wII,1}$ are more heterogeneous than the ones in $G^{wII,2}$. Most likely the reason of it is an absence in most cases of necessity to redistribute time for supporting the contacts in $G^{wII,1}$. Comparing the quality of the MLCD in $G^{wII,1}$, $G^{wII,2}$ with G^{wII} described in Section 5.2 we can conclude that: a) weights of the auxiliary networks $\{G^k\}_k$ are $(\omega(G^k))_k = (0.772, 1.330, 0.962)$, hence they are all not normalised and G^2 is “haviest”;

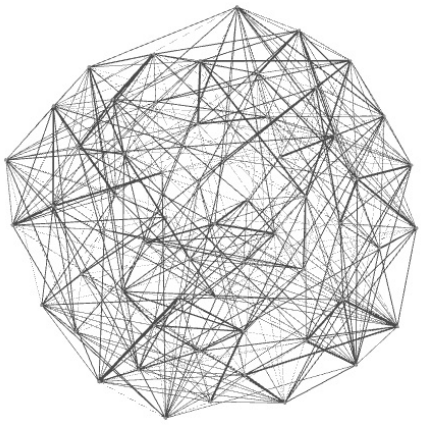


Figure 5.18: The HCN $G^{wII,1}$

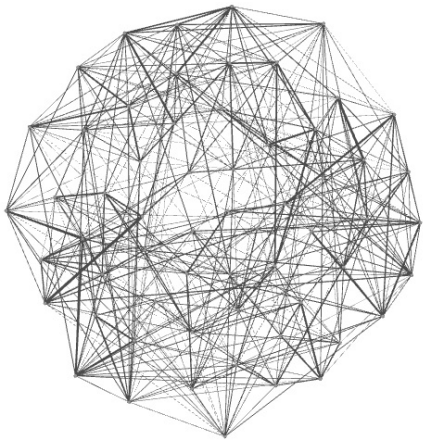


Figure 5.19: The HCN $G^{wII,2}$

b) in the HCNs the dominant node attribute, AT^{n2}, was detected in the original network without complementing it by an association network G^a (it corresponds the choice $(W^I, W'^I) = (0, 1)$). After the first iteration of the MLCD by (4.48) the weight of the network is decreased almost twice - $\omega(G(2)) = 1 - W^{II,2}\omega(G^2) = 1 - 0.443 = 0.557$; c) quality of the partition $\mathcal{AC}^2$ detection is never 100%. Namely, in 80% of cases two ACs in $\mathcal{AC}^2$ are detected correct, rest two with 1 error each in $G^{wII,1}$ (see Figure 5.20) and 3 errors each in $G^{wII,2}$ (see Figure 5.21).

Supposedly the reason of worst CD in $G^{wII,2}$ is more homogeneous edge weights and higher dependency between the three layers in $G^{wII,2}$ than in $G^{wII,1}$.

Conclusion. The HCN model is an example of attributed networks that are multi-layer networks and the layers are interconnected. Due to this difficulty, the MLCD and MLNI algorithms implementation could be complicated in

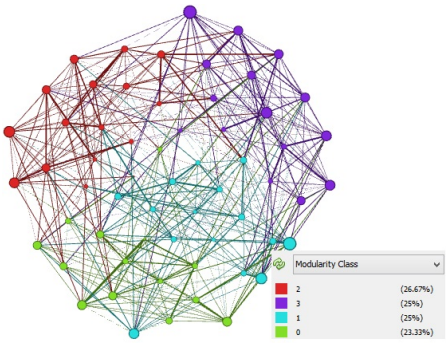


Figure 5.20: Communities in $G^{wII,1}$ ($M = 0.368$)

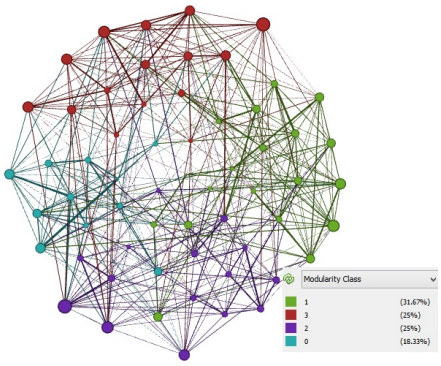


Figure 5.21: Communities in $G^{wII,2}$ ($M = 0.366$)

general HCNs. On the other hand, the restriction on communication time is natural and is used by most of people. Also notice that the restriction on communication time can be formulated as a restriction on capacity of nodes, hence the HCN model can be generalised for any network that have similar restrictions. Moreover it can be successfully applied to unweighted networks if weights are assigned as described above.

5.4 The HSTN Analysis

On this stage of the research we analyse the original High School Texting Network G as well as its robust subnetwork G' (see Section 4.2). To both of them we refer as the HSTN, but for the robust network we denote the corresponding networks by the mark $'$. For example, $G^{a'}, G^{e'}$ - are an association and weighted network corresponding to G', respectively.

5.4.1 The HSTN Social Network Properties

Consider the original HSTN G presented in Section 4.2 and analyse its social networks properties. We defined social network (see Sect.2.2) as people relationships networks. To determine whether a particular network graph is a social network is possible through numerical networks characteristics analysis. The HSTN is a network of texting contacts, not people relationships. Nevertheless we believe that texting communication reflect qualitatively these relationships, so, the HSTN behaves as social.

To accomplish the analysis the key numerical characteristics of G (see Section 2.2) are compared with the network $H = ERRG(n, p)$ corresponding ones, $p = \frac{2m}{n(n-1)}$ (see Table 5.9). Both networks have low densities, thus they are sparse ($\delta \approx 1.5\%$). Erdös-Rényi Random Graphs [ER59] have small diameter, small average shortest path and low average clustering coefficient. The diameter and the average shortest path in G are compatible with the H ($Diam = 8$, $\bar{l} = 3.841$) whiles the average clustering coefficient is noticeably higher ($\overline{CC} = 0.257$).

Remark 11. *Since [UKBM11] reports that the United States-users Facebook network had a diameter of 10, the value* $Diam = 8$ *seems enormous for such small network like the HSTN. Nevertheless it is true, nodes with several hops distance among the students of 9-th and 12-th grades can be easily seen in the Figure 5.24. Moreover, a diameter in the robust subnetwork* G' *even higher* ($Diam(G') = 9$)*. We believe that it is due to the students of different grades*

communicate, but primarily not via texting.

In Figure 5.22 we can see that the HSTN degree distribution is not perfect approximated by the power function, especially its left tail. The hypothesis about a power-law degree distribution is confirmed only with low probability. Overall, the HSTN demonstrated most of the listed social networks properties. In our opinion, the difference between a social network and the HSTN is not surprising as: a) the last one is a small-scale network; b) some of the high-school students did not participate in the survey; c) some inaccuracies occurred during the data collection.

i	statistics	I:G	II:ERRG	I/II
1	n	521	521	
2	m	1887	1995	
3	δ	0.014	0.015	0.946
4	$Diam$	8	5	1.6
5	$\bar{l}$	3.841	2.953	1.301
6	$\overline{CC}$	0.257	0.019	13.526

Table 5.9: The key social network's characteristics of the HSTN

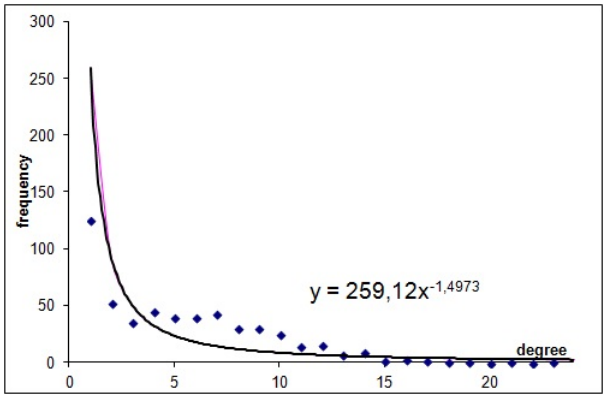


Figure 5.22: The HSTN degree distribution approximation by power function

5.4.2 Extracting Data Parameters from the HSTN

Edges Decoration by Weights The original HSTN G is unweighted and has the ranking edge attribute "TC" (see Section 4.2). We derive edge weights from G in order to maximize modularity. For that we follow the scheme described in Section 4.3.1. Series of 5 CD attempts were conducted and the average modularity $\overline{M}$ along with the highest modularity, M^{max}, were found. The only edge attribute "TC" takes values $\overline{at}^e = \{1, 2, 3\}$, so, $L' = 3$. The function (4.29) is obtained in the way described below:

1. First we use Scheme 1 described in Remark 7:

   - in the unknown vector $\overline{at}^{we}$ we fix $at_3^{we} = 1$, choose $at_1^{we} < at_2^{we} < 1$ and run CD. Results of the first trial with the step 0.2 are presented in Table 5.10. As we see $M^{max1} = \max_{i,j} \overline{M} = 0.638$ is attained for $(at_1^{we}, at_2^{we}) = (0.4, 0.6)$ and $\overline{at}^{we1} = (0.4, 0.6, 1)$;
   - we specify the weights in the second attempt with the step 0.1 around the point $(0.4, 0.6)$ (see Table 5.11) $M^{max2} = \max_{i,j} \overline{M} = 0.645$ is reached for $(at_1^{we}, at_2^{we}) = (0.3, 0.5)$ and $\overline{at}^{we2} = (0.3, 0.5, 1)$.

2. Secondly, we apply Scheme 2 from Remark 7:

   - we approximate the corresponding points $(at_l^e, at_l^{we})_l = ((1, 0.3), (2, 0.5), (3, 1))$ by a two-parameter regression $y = a^2 g(a^3 x)$. The best fits the exponential function $y^0 = 0.1594 \cdot e^{0.602x}$ (see Figure 5.23) therefore we take function (4.29) in the form $y = a^1 + a^2 e^{a^3 x}$ adding

just a shift to y^0. Using the current notations the function y^0 has parameters $\overline{a}^0 = (a^{1,0}, a^{2,0}, a^{3,0}) = (0, 0.1594, 0.602)$;

- we take parameters a^2, a^3 near $\overline{a}^0$, derive an a^1 from the condition $\varphi(L') = \varphi(3) = at_3^{we} = 1$ and apply CD. From Table 5.12 we see that $(a^2, a^3) = (0.17, 0.65)$ and $a^1 = -0.195$, respectively, it gives a new improvement of modularity $M^{max3} = \max_{i,j} \overline{M} = 0.669$. So, the final choice of the function (4.29) is $\varphi(x) = 0.17 \cdot e^{0.65x} - 0.195$, wherefrom $\overline{at}^{we3} = (at_i^{we})_{i \in J_3} = \{\varphi(l)\}_l = (0.131, 0.429, 1)$. Along with the final weights choice there are three $\{M^{max,i}\}_{i \in J_3}$ values and corresponding weights $\{\overline{at}^{we,i}\}_{i \in J_3}$ are presented in Table 5.13.

$at_i^{we} \backslash at_j^{we}$	0.2	0.4	0.6	0.8	average
0.2		0.629	0.633	0.636	0.633
0.4	0.629		**0.638**	0.636	0.634
0.6	0.633	**0.638**		0.627	0.632
0.8	0.636	0.636	0.627		0.633
average	0.633	0.634	0.632	0.633	0.638

Table 5.10: Scheme 1 - $\overline{M}$ (at_i^{we}-step 0.2)

$at_i^{we} \backslash at_j^{we}$	0.3	0.4	0.5	0.6	average
0.3		0.641	**0.645**	0.636	0.641
0.4	0.641		0.639	0.638	0.639
0.5	**0.645**	0.639		0.635	0.64
0.6	0.636	0.638	0.635		0.636
average	0.641	0.639	0.64	0.636	0.645

Table 5.11: Scheme 1: $\overline{M}$ (at_i^{we}-step 0.1)

The last step is the normalization (4.32): $\overline{W}'^{III} = \frac{(0.131,0.429,1)}{||(0.131,0.429,1)||} = \frac{1}{1.56}(0.131, 0.429, 1) = (0.084, 0.275, 0.641)$. The vector $\overline{W}^{III}$ we use in further calculations as a set of edge weights. It is also weights of the "TC"-ranking

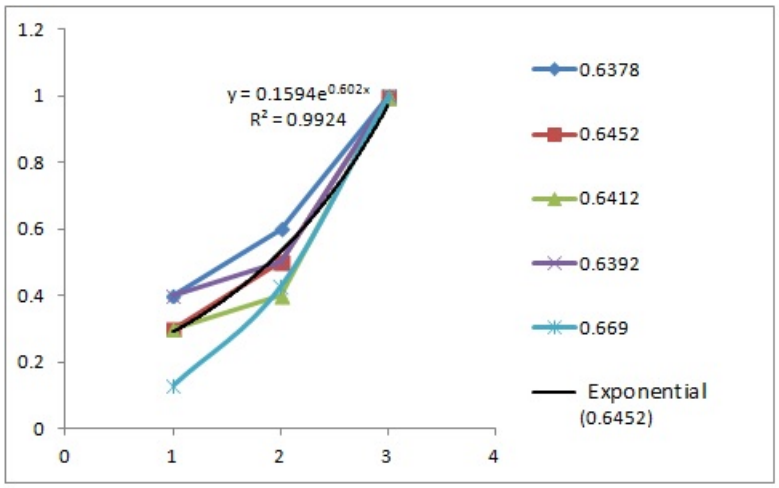


Figure 5.23: $\overline{at}^{we2}$ approximation by exponential function $y^0(x)$

values. The proportion $W_1'^{III} : W_2'^{III} : W_3'^{III} = 1 : 3.28 : 2.33$ can be interpreted in the following manner: in the HSTN community structure "worm" texting contacts are 3.28 times more important than "cold" contacts, in turn "hot" ones are 2.33 times more important than "worm". Thus, a "worm" rank for the "TC" is closer to a "hot" one rather than to a "cold" one.

The result of the step is assigning edge weights in G and its converting into the weighted network G^w. The same weights assigning is applied to G' and the robust weighted HSTN subnetwork G'^w is obtained. Further we will analyse both these weighted networks.

Extracting Weights of Node Attributes from the HSTN To evaluate a tuple $\overline{AT}^{II}$ of node attribute priorities we apply (4.27) and (4.28) to the network G^w (see Table 5.14). As we see both vectors $\overline{W}_1^{II}$, $\overline{W}_2^{II}$ are very

$a^2 \backslash a^3$	0.55	0.6	0.65
0.15	0.631	0.64	0.659
0.16	0.632	0.641	0.655
0.17	0.645	0.638	**0.669**

Table 5.12: Scheme 2 - $\overline{M}$ (a^2, a^3-step 0.05)

$\overline{at}^{wei}$	at_1^{wei}	at_2^{wei}	at_3^{wei}	M^{max}
$\overline{at}^{we1}$	0.4	0.6	1	0.648
$\overline{at}^{we2}$	0.3	0.5	1	0.647
$\overline{at}^{we3}$	**0.131**	**0.429**	**1**	**0.672**

Table 5.13: Scheme 2 - the final $\overline{at}^{we}$ choice

AT^{nk}	$d^{\mathcal{AC}^k}$	$d^{int,\mathcal{AC}^k}$	$W_1^{II,k}$	$\omega(\mathcal{AC}^k)$	$\omega^{int}(\mathcal{AC}^k)$	$W_2^{II,k}$	rank
Gender	1770	774	0.22	566.56	210.77	0.22	2
Sports	1214	1408	0.14	356.52	447.12	0.13	4
Science	996	1624	0.12	314.87	488.62	0.12	6
Gaming	1178	1442	0.14	378.06	425.42	0.14	3
Region	1064	1428	0.13	329.87	438.2	0.13	5
Grade	2054	450	0.25	647.77	120.92	0.25	1
Total			1			1	

Table 5.14: Node attribute weights assessment

similar and they specify an order of the node attributes from the strongest "Grade" to the weakest "Region". For further computations we choose $\overline{W}^{II} = \overline{W}_2^{II} = (W_2^{II,k})_k = (0.22, 0.13, 0.12, 0.14, 0.13, 0.25)$.

5.4.3 Problem 3 in the HSTN G'

Consider the robust HSTN G' and study nature of communities of their first layer, which are obtained by CD applied to G' directly. Since all the node attributes are known we can compose G'^w with the corresponding association network (see Section 4.3.1) denoted by $G^{a'}$. Assuming that $\overline{AT}^n$ are only reasons of the attributed network stricture formation we choose equal weights of the attributed and structural parts ($W^I = W^I = 0.5$) and construct an aggregated attributed network $G^{wa'} = 0.5G^{a'} + 0.5G^{a'}$. We conduct CD in the

networks $G^{w'}$ and $G^{wa'}$, obtain the node partitions $\mathcal{C}^{*'}$ and $\mathcal{C}^{*'a}$, respectively, and compare the results (see Figures 5.24-5.25):

- the partition $\mathcal{C}^{*'}$ includes $L^{*'} = 25$ communities. First eight, $C^{*'(0)} - C^{*'(7)}$ are the largest ones. They contain at least 2% each of the total number of the nodes and can be clearly seen in Figure 5.24. The rest 17 communities - we denote them $\mathcal{C}^{*'(8-24)}$ - primarily correspond to people did not provided their texting contacts therefore represented by isolated vertices or by communities with a few nodes. In terms of the HSTN, since only the 8-11-th grades high-school students participated in the survey, it is unlikely that they do not communicate via texting or by another way with someone of classmates, hence existence of these isolated components is explained exclusively by lost edges.

- For the network $G^{wa'}$ the situation looks different. The network is one-component, the partition $\mathcal{C}^{*'a}$ includes only $L^{*'a} = 8$ communities, where first seven contain all nodes except for one. Namely these communities denoted by $\mathcal{C}^{*'a(0-6)}$ we analyse. Notice that they can be clearly seen in Figure 5.25.

These pictures demonstrate an advantage of the usage of network aggregation with an associated network instead of the original network usage. In the aggregated HSTN almost all the mentioned isolated components became connected by common node attributes edges. As a result the number of communities decreased significantly and made possible the CD results interpretation. Also,

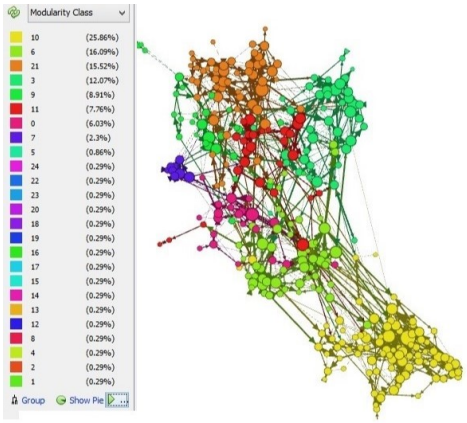


Figure 5.24: Communities in the HSTN $G^{w'}$ ($M = 0.646$)

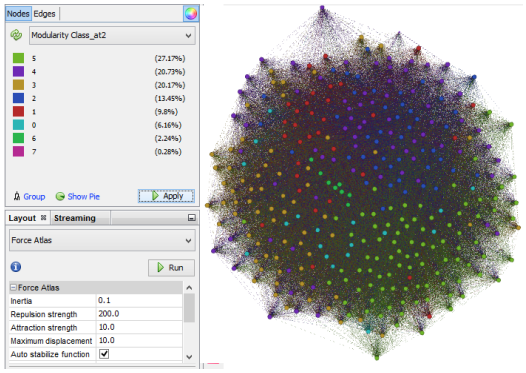


Figure 5.25: Communities in the aggregated $G^{wa'}$ ($M = 0.314$)

since the number of isolated components decreases a lot it makes possible to restore all the missing network information: we can start with node and edge attributes inference, after that the edge inference of the corresponding associated network can be conducted based on already renovated node attributes (see Section 3.1). In terms of the HSTN, it means that the applying the aggregated networks allows for any node with at least one link or a node attribute to restore the rest of attributes, whiles for the original network it is not possible. For the HSTN community structure studying the aggregated network provides an opportunity to clarify to which communities the students from the $\mathcal{C}^{*'(8-24)}$ belong if their links to the community were lost during the initial data collecting.

We analyse communities in $\mathcal{C}^{*'a(0-6)}$ and assign underlying ATNVs (UATNVs) (see (4.58)) to them with the level of significance $\alpha = 0.1$ according to the scheme described in Section 4.3.2. The result presented in Tables 5.15-5.16 is

obtained for those four node attributes, for which the communities UATNVs were derived. The other two node attributes, "Science" and "Region", are not significant in the division $\mathcal{C}^{*'}$. There are evaluated proportions (4.55) and (4.56) in Table 5.15. In Table 5.16 these underlying communities attribute values (4.58) for $\alpha = 0.1$ are presented. Notice that for each of the communities all the justified "Grade"-values were extracted and AT^{n6} ="Grade" is the dominant node attribute of the partition ($k^{**} = k^* = 6$ in (4.63),(4.65)), as well for the original G, G' and the accumulated G^{wa}.

AT^{nk}		Gender		Sports			Gaming			Grade			
l^*	n_{l^*}	f	m	1	2	3	1	2	3	9	10	11	12
5	95	40%	60%	20%	18%	62%	35%	33%	33%	89%	2%	7%	1%
4	73	60%	40%	33%	19%	48%	55%	18%	27%		8%	5%	86%
3	71	59%	41%	23%	28%	49%	39%	30%	31%	1%	69%	14%	15%
2	47	38%	62%	15%	19%	66%	51%	19%	30%	2%		91%	6%
1	34	79%	21%	15%	35%	50%	38%	47%	15%	3%		88%	9%
0	19	63%	37%	21%	16%	63%	47%	37%	16%	11%	84%	5%	
6	8	88%	13%	25%	25%	50%	75%		25%			100%	
p_l^{0k}		54%	46%	22%	22%	56%	44%	28%	28%	26%	21%	30%	23%

Table 5.15: Justification of the $\mathcal{C}^{*'a(0-7)}$-communities attributes. Proportions p_l^{0k} and $p_{ll^*}^{k}$ comparison

l^*	n_l	Gender	Sports	Gaming	Grade
5	95	male			9
4	73		1	1	12
3	71				10
2	47	male			11
1	34	female	2	2	11
0	19				10
6	8	female		1	11

Table 5.16: Underlying $\mathcal{C}^{*'a(0-6)}$-communities attributes, $p = 0.9$

For example, review the communities $C_1^*, C_2^*, C_6^* \in \mathcal{C}^{*'a(0-6)}$ of the Grade 11.

There are only one "male" community, C_2^*, and two "female" communities: the C_1^* unites girls that like sport and gaming activities, but they are not fans of it; there are mostly girls who definitely do not like gaming in the C_6^*. Two communities relates to the Grade 10, $C_0^*, C_3^* \in \mathcal{C}^{*\prime a(0-6)}$. For them no underlying attributes among $AT^{n1} - AT^{n6}$ were derived, hence with the probability 90% the reason of this division remains undetected. It implies that likely we can obtain UATNVs and UATNs with a lower probability or the division is caused by some the HSTN features that were not included in $\overline{AT}^n$. Therefore the study is not complete.

Conclusion. It was shown how to assign underlying attributes to communities, justify the choice, and interpret the results. Additionally we conducted $\tau = 1$-iteration of the MLCD algorithm in $G^{wa'}$ and started Problem 1 solving in G^w.

Notice that the node attribute "Grade" is in fact dominant. Identical results about dominance of the attribute were obtained for the original network G^w as well as for the aggregated networks G^{wa}, $G^{wa'}$. This is not surprising because the attribute "Grade" has the greatest weight $W^{II,6} = 0.25$.

5.4.4 Problem 2 in the HSTN G^w

The MLNI algorithm is implemented to the incomplete attributed HSTN G^w by following the scheme used for the network G^{wII} in Section 5.2. Statistics of incompleteness is shown in Table 4.1. Because of it the aggregated network G^w can be complemented only by an incomplete association network G^{0a}, $G^{0wa} =$

$0.5G^{0a} + 0.5G^{wII}$. We conduct CD on $G(1) = G^{0wa}$ and obtain the partition $\mathcal{C}^1$ that is shown in Figure 5.26. There are only 6 main communities, which contain at least 5% of nodes each, in the $\mathcal{C}^1$. Among these six communities, similarly to $G^{wa'}$ (see Section 5.4.3, Table 5.16), two of them correspond to the Grades 9 and 12 and the rest four - to the Grades 10 and 11 (two communities in each). In Figure 5.26 we can see that the communities are allocated. The node attribute underlying $\mathcal{C}^1$ is again AT^{n6}, which is also dominant, $p = 90\%$. In Table 5.17 we can see how the Grades of 156 high-school students were restored. We reduce weights by extracting the network $G^{0wa,6}$

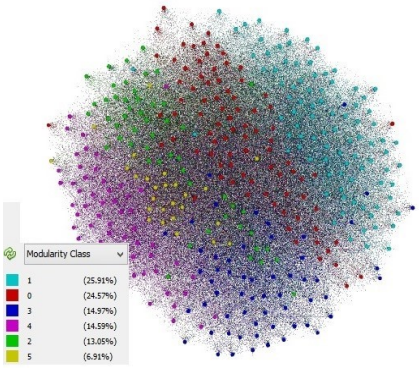


Figure 5.26: Communities in the HSTN $G(1) = G^{w}$ $(M = 0.362)$

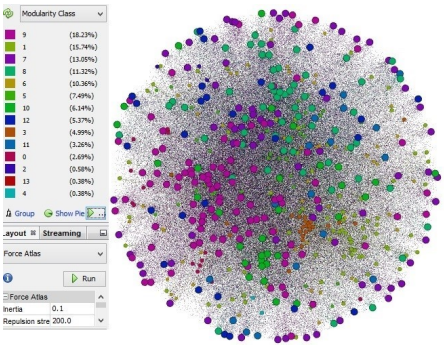


Figure 5.27: Communities in the HSTN, iteration $\tau = 2$, $G(2)$ $(M = 0.744)$

from $G(1)$ and conduct $\tau = 2$ step of the MLNI for $G(2) = G(1) - G^{0wa,6}$. The result is the partition $\mathcal{C}^2$ presented in Figure 5.27. It is clear that the dominant community partition related to Grades is destroyed and the shown new community structure is entirely different. The likely underlying attribute of $\mathcal{C}^2$ is "Gender", which has the second greatest weight $W^{II,1} = 0.22$, but

l	Grade	initial incomplete data	restored data	total
0	0	156		
1	9	93	43	136
2	10	78	63	141
3	11	109	50	159
4	12	85	0	85
Total		365	156	521
%		70.1%	29.9%	100%

Table 5.17: Restoring ATNVs related to AT^{n6}

this choice with the chosen probability is not justified.

Similar results were obtained during Problem 1 solving on G^w and performing further iterations of the MLCD in $G^{wa'}$ (see Section 5.4.3).

Conclusion. It was found that in the HSTN the main reason of dividing into communities within the high school is "Grade"-division. When the initial HSTN G^w was accompanied by the association network, we were able to restore 30% of the missing "Grade"-node attributes and justify the result with probability 90%. We succeeded to reduce the impact of the dominant "Grade"-partition that was enough for detecting community structure of the next level of the network.

Chapter 6

Conclusion

In the scope of the thesis it was proposed to decompose an attributed network into subnetworks related to single node attributes. For utilization of all available network information we constructed an aggregated network from the initial network and its corresponding association network. We used the aggregated network for multi-layer community detection and nodes' attributes restoration. We presented iterative algorithms based on the detection of the underlying attributes of node partitions and extracting the correspondent subnetworks from the consideration. We introduced a number of models for attributed networks formation, which are based on the famous Erdös-Rényi and Barabási-Albert random graph models. In particular, we formulated the Human Communication Network model with the restriction on a communication time. We introduced the High School Texting Network and used it for the demonstration of the performance of our approaches. It was also

demonstrated that the proposed attributed network formation models better simulate social networks than their underlying random graph models. Our approach essentially uses weights of the network components – edges, sub-networks, nodes' and edges' attributes, etc. In order to reduce the cost for derivation of the additional information, e.g. expert assessment, we used our own technique that allows extracting a part of the information directly from the network.

The research can be extended in several directions: a) the technique that we used for association networks and the Human Communication Network of converting the unweighted networks into the weighted one based on node attributes, few parameters and assumptions can be also applied for other networks; b) the multi-layer CD approach used for incomplete information attributed networks can be extended by seed-based CD algorithms; c) the Multi-Layer Node Attribute Inference algorithm can be extended and applied for the attributed network, which does not have any information about node attributes; d) for choosing underlying attributes of node partitions statistics based on network structural characteristics, such as intra- and inter-cluster densities, can be used.

Bibliography

[AB02] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.

[BC09] M. J. Barber and J. W. Clark. Detecting network communities by propagating labels under constraints. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 80(2 Pt 2):026129:1–026129:16, August 2009.

[BDG+08] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, (2):172–188, 2008.

[BGL05] J. Brüning, V. A. Geiler, and I. S. Lobanov. Spectral Properties of Schrodinger Operators on Decorated Graphs. *Mathematical*

Notes, 77(5/6):858–861, June 2005.

[BGLL08] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory & Experiment*, 2008(10):1–12, October 2008.

[BHS13] B. Boden, R. Haag, and T. Seidl. Detecting and exploring clusters in attributed graphs. In *Conference on Information & Knowledge Management*, pages 2505–2508, January 2013.

[CA12] E. Cuvelier and M.-A. Aufaure. Graph Mining and Communities Detection. In *Business Intelligence*, number 96 in Lecture Notes in Business Information Processing, pages 117–138. Springer Berlin Heidelberg, January 2012.

[CMN08] A. Clauset, C. Moore, and M.E.J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, (7191):98–101, 2008.

[CNM04] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 70(6 Pt 2):066111:1–066111:6, December 2004.

[Das14] B. DasGupta. Computational Complexities of Optimization Problems Related to Model-Based Clustering of Networks. In

Themistocles M. Rassias, Christodoulos A. Floudas, and Sergiy Butenko, editors, *Optimization in Science and Engineering*, pages 97–113. Springer New York, January 2014.

[ER59] P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[For10] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, January 2010.

[GN02] M. Girvan and M. E. J. Newman. Community Structure in Social and Biological Networks. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 99/12, pages 7821–7826, June 2002.

[Gre10] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):1–26, October 2010.

[HM14] R. G. Hitesh and V. D. Maulik. A Survey on Community Detection in Weighted Social Network. *International Journal of Advance Research in Computer Science and Management Studies*, 2(1):474–479, January 2014.

[JPWX11] J. Jin, L. Pan, C. Wang, and J. Xie. A Center-Based Community Detection Method in Weighted Networks. In *2011 23rd IEEE*

International Conference on Tools with Artificial Intelligence (ICTAI), pages 513–518, January 2011.

[KK14] I. M. Kloumann and J. M. Kleinberg. Community Membership Identification from Small Seed Sets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1366–1375, New York, NY, USA, 2014. ACM.

[KL11] M. Kimand and J. Leskovec. The Network Completion Problem: Inferring Missing Nodes and Edges in Networks. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 47–58, SIAM, 2011. Omnipress.

[Kol09] E. D. Kolaczyk. *Statistical Analysis of Network Data*. Springer Series in Statistics. Springer New York, New York, NY, 2009.

[KPS13] K. Kothapalli, S.V. Pemmaraju, and V. Sardeshmukh. On the Analysis of a Label Propagation Algorithm for Community Detection. In S. Sarkar R. K. Shyamasundar D. Frey, M. Raynal and P. Sinha, editors, *Distributed Computing and Networking*, number 7730 in Lecture Notes in Computer Science, pages 255–269. Springer Berlin Heidelberg, January 2013.

[LF09] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Physical Review. E, Statisti-*

cal, Nonlinear, And Soft Matter Physics, 80(5 Pt 2):056117:01–056117:17, November 2009.

[Li13] J. Li. Detecting overlapping communities by seed community in weighted complex networks. *Physica A*, 392(23):6125–6134, 2013.

[LKY+12] W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li. Community Detection in Incomplete Information Networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 341–350, New York, NY, USA, 2012. ACM.

[LLM10] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical Comparison of Algorithms for Network Community Detection. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 631–640, New York, NY, USA, 2010. ACM.

[LM10] X. Liu and T. Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389:1493–1500, January 2010.

[LWBC04] R. J. Wilson L. W. Beineke and P. J. Cameron, editors. *Topics in Algebraic Graph Theory*. Cambridge University Press, Cambridge, UK ; New York, 1 edition edition, October 2004.

[LWC13] Z. Lu, Y. Wen, and G. Cao. Community detection in weighted networks: Algorithms and applications. In *2013 IEEE International Conference on Pervasive Computing & Communications (PerCom)*, pages 179–184, January 2013.

[LZXC14] Z. Lin, X. Zheng, N. Xin, and D. Chen. CK-LPA: Efficient community detection algorithm based on label propagation with community kernel. *Physica A: Statistical Mechanics and its Applications*, 416:386–399, December 2014.

[New03] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, January 2003.

[New04a] M. E. J. Newman. Analysis of weighted networks. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 70(5 Pt 2):056131–056131, November 2004.

[New04b] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 69(6 Pt 2):066133:1–066133:5, June 2004.

[New06a] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 74(3 Pt 2):036104:1–036104:22, September 2006.

[New06b] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June 2006.

[New12] M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, January 2012.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 69(2 Pt 2):026113:1–026113:16, February 2004.

[PDFV05] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[RAK07] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 76(3 Pt 2):036106:1–036106:12, September 2007.

[RB06] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 74(1 Pt 2):016110:1–016110:16, July 2006.

[RN09] P. Ronhovde and Z. Nussinov. Multiresolution community detection for megascale networks by information-based replica

correlations. *Physical Review. E, Statistical, Nonlinear, And Soft Matter Physics*, 80(1 Pt 2):016109:1–016109:19, July 2009.

[RSM+02] E. Ravasz, A. L. Somera, D. A. Mongruand, Z. N. Oltvai, and A.-L. Barabási. Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297(5586):1551–1555, August 2002.

[SA89] T. L. Saaty and J. M. Alexander. *Conflict Resolution: The Analytic Hierachy Approach*. Praeger Pub, New York, October 1989.

[Saa77] T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281, 1977.

[Sch07] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, August 2007.

[SMM14] C.L. Staudt, Y. Marrakchi, and H. Meyerhenke. Detecting communities around seed nodes in complex networks. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 62–69, October 2014.

[TB09] V. A. Traag and J. Bruggeman. Community detection in networks with positive and negative links. *Physical Review. E, Sta-*

tistical, Nonlinear, And Soft Matter Physics, 80(3 Pt 2):036115–036115, September 2009.

[THP08] Y. Tian, R. A. Hankins, and J. M. Patel. Efficient Aggregation for Graph Summarization. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 567–580, New York, NY, USA, 2008. ACM.

[UKBM11] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. 2011.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[XKW+12] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng. A Model-based Approach to Attributed Graph Clustering. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 505–516, New York, NY, USA, 2012. ACM.

[YGFG+05] P. Yolum, T. Güngör, C. Özturan F. Gürgen, P. Pons, and M. Latapy. Computing Communities in Large Networks Using Random Walks. In *Computer & Information Sciences - ISCIS 2005*, pages 284–293. January 2005.

[YL12] J. Yang and J. Leskovec. Defining and Evaluating Network Communities Based on Ground-truth. In *Proceedings of the*

ACM SIGKDD Workshop on Mining Data Semantics, MDS '12, pages 3:1–3:8, New York, NY, USA, 2012. ACM.

[YL13] J. Yang and J. Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 587–596. ACM, 2013.

[YML13] J. Yang, J. McAuley, and J. Leskovec. Community Detection in Networks with Node Attributes. In *2013 IEEE 13th International Conference on Data Mining (ICDM)*, pages 1151–1156, December 2013.

[ZCY09] Y. Zhou, H. Cheng, and J. X. Yu. Graph Clustering Based on Structural/Attribute Similarities. *Proc. VLDB Endow.*, 2(1):718–729, August 2009.

[ZCY10] Y. Zhou, H. Cheng, and J.X. Yu. Clustering Large Attributed Graphs: An Efficient Incremental Approach. In *2010 IEEE 10th International Conference on Data Mining (ICDM)*, pages 689–698, December 2010.

Appendices

Appendix A

List of Abbreviations

ACC	average clustering coefficient
AHP	Analytic Hierarchy Process
AI(s)	activity/interest (activities/interests)
ANCDA(s)	community detection algorithm(s) for attributed networks
ASPL	average shortest path length
ATEs	edges' attributes
ATEVs	edges' attribute values
ATN(s)	discrete nodes' attribute(s)

ATNVs	discrete nodes' attribute values
CD	community detection
CDA(s)	community detection algorithm(s)
CIAN/IIAN	complete /incomplete information attributed network
GTCs	ground-truth communities
HCN	Human Communication Network
LATN(s)	likely underlying node attribute(s)
LATNVs	likely underlying node attribute value(s)
LPA	Label Propagation Algorithm
MLCD	Multi-layer Community Detection
MLNI	Multi-layer Node Attribute Inference
SCDP	Selective Community Detection Problem
SECP	Seed Sets Expansion Coverage Problem
SNEP	Seed Nodes Expansion Problem
SEPP	Seed Sets Expansion Partition Problem
SSEP	Seed Set Expansion Problem
UATN(s)	underlying node attribute(s)

UATNV(s) underlying node attribute value(s)

WAM(s) weighted adjacency matrix/matrices